



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Interaction Concepts for Knowledge Work on Augmented Desktops

Dissertation

zur Erlangung des akademischen Grades
eines Doktor rerum naturalium (Dr. rer. nat.)

Eingereicht von

Jan Christoph Riemann, M.Sc.

geboren in Darmstadt

Angenommen vom

Fachbereich Informatik der Technischen Universität Darmstadt

Gutachter: Prof. Dr. Max Mühlhäuser
(Technische Universität Darmstadt)
Prof. Dr. Jan Borchers
(RWTH Aachen)

Tag der Einreichung: 06. November 2018

Tag der Disputation: 20. Dezember 2018

Darmstadt 2018

Interaction Concepts for Knowledge Work on Augmented Desktops

Genehmigte Dissertation von Jan Christoph Riemann, M.Sc.

Darmstadt, Fachbereich Informatik, Technische Universität Darmstadt

1. Gutachten: Prof. Dr. Max Mühlhäuser
2. Gutachten: Prof. Dr. Jan Borchers

Tag der Einreichung: 06. November 2018

Tag der Disputation: 20. Dezember 2018

Jahr der Veröffentlichung der Dissertation auf TUpriints: 2019

Alle Rechte vorbehalten.

Wiedergabe oder auszugsweise Veröffentlichung nur mit Genehmigung des Autors.

© 2018

Abstract

With the proliferation of computers in today’s workplaces, work processes have changed: Especially in the field of knowledge work, which heavily relies on the use of information sources, electronic documents have become commonplace. However, despite the advantages of digital systems, like global access to information or advanced search facilities, physical media have not yet vanished. The amount of paper used is even still increasing. Even digitally available documents are often printed in order to leverage the unique affordances of printed paper (e.g., its flexibility, low weight, etc.). Another particular advantage of physical media on physical desks is their aptitude to support the spatial layout of sets of documents (e.g., piling or spreading them out or assigning categories to them through their spatial position, e.g., by placing them in the storage area vs. working area, or even more specialized areas on the desk for categories like “to be read” or “urgent”). As a result, most work processes do not only require the use of either digital or physical documents but the concurrent use of both types. We use the term “hybrid” to denote this concurrent use.

As of today, there is still only quite limited support for work practices that encompass digital and physical documents concurrently. According to common practice at stationary workplaces, digital documents are stored on a computer and viewed on a monitor in front of the user, while physical documents are placed on a desk, leading to a strict separation of digital and physical documents.

For several years, so-called digital tabletops have been available: desks with a large flat display acting as the desk surface, supplemented with touch-sensing capabilities. In principle, it is now possible for digital and physical documents to coexist on the table surface, laying the foundation for more sophisticated working practices. However, more than a decade since the advent of tabletops, the integration of digital and physical knowledge work still leaves a lot to be desired. Where digital tabletops are used today, their surface is always kept entirely clean of physical objects so as not to obstruct any of the interactive display surface. Since this is a quite alien use pattern for desks, tabletops have not yet gained a reasonable market share.

The primary goal of this thesis is to foster tighter integration between the digital and physical worlds. To do so, it first contributes the *PeriTop* concept, a two-class display concept that leverages the surfaces of objects as a secondary

peripheral display in addition to the primary tabletop display. PeriTop serves as a basis for several further contributions presented in this thesis. All further contributions together are categorized into three main areas, namely, *stationary hybrid knowledge work*, *layout for hybrid environments*, and *mobile workspaces*.

In the first area, *stationary hybrid knowledge work*, this thesis contributes (1) *ProjecTop*, a novel concept for leveraging the surface of occluding objects to resolve occlusion “in situ” without the need for using additional display space on the tabletop, and (2) the *StackTop* concept for hybrid stacking, which allows arbitrary interweaving of digital and physical documents in a coherent stack.

In the second area, *layout for hybrid environments*, this thesis contributes (1) *FreeTop*, a flexible approach for assessing projection quality on surfaces in order to facilitate projection-based augmentation of physical objects, and (2) the *FlowPut* concept, an environment-aware layout system for interactive tangibles based on FreeTop.

In the third area, *mobile workspaces*, the aspect of mobile work in the context of hybrid work environments is tackled by contributing *PiraTop*, a concept for augmented reality (AR)-based remote access to stationary hybrid workspaces. It leverages the idea of spatial memory to facilitate access.

The final contribution of this thesis is the *OverTop* concept. OverTop combines the advantage of AR and the knowledge gained with PiraTop with the contributions on stationary desks, particularly PeriTop. Therefore, it considers head-mounted augmented reality displays as an addition to stationary tabletops, making the space above the desk surface and above the objects that populate the desk available for hybrid knowledge work.

The thesis concludes with a summary and an outlook on future work directions.

Zusammenfassung

Durch die Verbreitung von Computern an heutigen Arbeitsplätzen haben sich die Arbeitsprozesse verändert: Gerade im Bereich der Wissensarbeit, die stark auf die Nutzung von Informationsquellen angewiesen ist, sind elektronische Dokumente alltäglich geworden. Trotz der Vorteile digitaler Systeme, wie globaler Zugang zu Informationen oder erweiterte Suchmöglichkeiten, sind physische Medien jedoch noch nicht verschwunden. Im Gegenteil, der Papierverbrauch steigt sogar noch weiter an. Selbst digital verfügbare Dokumente werden oft gedruckt, um die Vorteile von bedrucktem Papier (z.B. Flexibilität, geringes Gewicht, etc.) zu nutzen. Ein weiterer Vorteil physischer Medien auf physischen Schreibtischen ist die Möglichkeit Dokumente räumlich anzuordnen (z.B. sie zu stapeln oder auszubreiten, sie – durch ihre räumliche Position – zu kategorisieren, z.B. durch Platzieren im Lagerbereich oder Arbeitsbereich des Schreibtischs, oder durch Nutzung noch weiter differenzierter Bereiche auf dem Schreibtisch für Kategorien wie „zu lesen“ oder „dringend“). Infolgedessen erfordern die meisten Arbeitsprozesse nicht nur die Verwendung von digitalen oder physischen Dokumenten, sondern die gleichzeitige Nutzung beider Dokumentarten. Diese gleichzeitige Nutzung digitaler und physischer Dokumente wird in dieser Arbeit als „hybrid“ bezeichnet.

Bis heute gibt es nur eine sehr begrenzte Unterstützung für Arbeitspraktiken, die digitale und physische Dokumente gleichzeitig umfassen. Üblicherweise werden digitale Dokumente auf einem Computer gespeichert und auf einem vor dem Benutzer stehenden Monitor betrachtet, während physische Dokumente auf dem Schreibtisch abgelegt werden, was zu einer strikten Trennung von digitalen und physischen Dokumenten führt.

Seit einigen Jahren gibt es so genannte digitale Tabletops: Tische mit einem großen Bildschirm mit integrierter Toucherkennung als Tischoberfläche. Prinzipiell ermöglichen solche Tabletops, dass digitale und physische Dokumente auf der Tischoberfläche koexistieren und schaffen somit die Grundlage für integriertere Arbeitsabläufe. Mehr als ein Jahrzehnt nach dem Aufkommen von Tabletops lässt die Integration von digitaler und physischer Wissensarbeit jedoch noch immer viel zu wünschen übrig. Werden heute Tabletops verwendet, wird ihre Oberfläche nach Möglichkeit immer völlig frei von physischen Gegenständen gehalten, um die interaktive Displayoberfläche nicht zu verdecken. Diese Nutzung steht im Widerspruch zur klassischen Nutzung von Schreib-

tischen, die grade dazu dienen, physische Gegenstände abzulegen. Als Folge davon haben Tabletops noch immer keinen größeren Marktanteil gewonnen.

Das Hauptziel dieser Arbeit ist es, eine engere Integration zwischen der digitalen und der physischen Welt zu fördern. Dazu trägt sie zunächst das *PeriTop* Konzept bei, ein Zweiklassen-Displaykonzept, das die Oberflächen von Objekten als sekundäre periphere Displays neben dem primären Tabletopdisplay nutzt. *PeriTop* dient als Grundlage für mehrere weitere Beiträge, die in dieser Arbeit vorgestellt werden. Alle weiteren Beiträge sind in drei Hauptbereiche unterteilt, nämlich *stationäre hybride Wissensarbeit*, *Layout für hybride Umgebungen* und *mobiles Arbeiten*.

Im ersten Bereich, der *stationären hybriden Wissensarbeit*, stellt diese Arbeit zwei Konzepte vor: (1) *ProjecTop*, ein neuartiges Konzept zur Nutzung der Oberfläche von physischen Objekten zur in situ Auflösung von Verdeckung ohne die Notwendigkeit der Nutzung zusätzlicher Displayfläche auf dem Tabletop, und (2) das *StackTop* Konzept für hybrides Stapeln, das eine beliebige Anordnung von digitalen und physischen Dokumenten in einem zusammenhängenden Stapel ermöglicht.

Im zweiten Bereich, *Layout für hybride Umgebungen*, stellt diese Arbeit ebenfalls zwei Konzepte vor: (1) *FreeTop*, einen flexiblen Ansatz zur Beurteilung der Projektionsqualität auf Oberflächen, um eine projektionsbasierte Augmentierung physikalischer Objekte zu ermöglichen, und (2) das *FlowPut*-Konzept, ein sich dynamisch an die Umgebung anpassendes Layout-System für interaktive Tabgibles auf Basis von *FreeTop*.

Im dritten Bereich, *mobiles Arbeiten*, wird der Aspekt der mobilen Arbeit im Kontext hybrider Arbeitsumgebungen betrachtet, indem *PiraTop*, ein Konzept für den Augmented Reality (AR)-basierten Fernzugriff auf stationäre hybride Arbeitsplätze, vorgestellt wird. *PiraTop* basiert auf der Idee der Nutzung des räumlichen Gedächtnisses, um den Zugang zu Informationen zu erleichtern.

Der letzte Beitrag dieser Arbeit ist das *OverTop* Konzept. *OverTop* kombiniert den Vorteil von AR und das mit *PiraTop* gewonnene Wissen mit den Beiträgen auf stationären Schreibtischen, insbesondere *PeriTop*. *OverTop* nutzt dazu Augmented Reality Brillen als Ergänzung zu stationären Tabletops, die den Raum über der Tischoberfläche und über den Objekten, die sich auf dem Schreibtisch befinden, für hybride Wissensarbeit nutzbar machen.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf weiterführende Forschungsthemen ab.

Erklärung zur Dissertation

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ein Promotionsversuch fand bisher nicht statt.

Darmstadt, den 06. November 2018

(Jan Riemann)

Contents

I	Context and Preliminaries	1
1	Introduction	3
1.1	Motivation and Context	3
1.2	Thesis Structure	11
1.3	Publications	12
2	PeriTop – Hybrid Tabletop Environment	15
2.1	Introduction	15
2.2	Related Work	16
2.3	Preliminaries	25
2.4	PeriTop	34
2.5	Use Case Scenarios	36
2.6	Implementation Issues	40
2.7	Summary and Conclusion	45
II	Stationary Hybrid Knowledge Work	47
3	ProjecTop – Resolving Occlusion in Situ	49
3.1	Introduction	49
3.2	Related Work	51
3.3	Initial Study	56
3.4	ProjecTop	61
3.5	User Study	67
3.6	Results and Discussion	71
3.7	Implications	76
3.8	Summary and Conclusion	78
4	StackTop – From Occlusion to Hybrid Stacking	79
4.1	Introduction	79
4.2	Related Work	81
4.3	StackTop	86
4.4	User Study	94
4.5	Results and Discussion	95
4.6	Implications for Hybrid Stacking	100
4.7	Summary and Conclusion	101

III	Layout for Hybrid Environments	103
5	FreeTop – Solving the Placement Problem	105
5.1	Introduction	105
5.2	Related Work	107
5.3	FreeTop Projectability Map	113
5.4	System Description and Experience	121
5.5	Summary and Conclusion	122
6	FlowPut – Embedding Tangibles	123
6.1	Introduction	124
6.2	Related Work	126
6.3	FlowPut Overview	134
6.4	Object Tracking and Touch Input Detection	136
6.5	Environment-aware UI Layout	143
6.6	Applicability	150
6.7	Implementation	152
6.8	Evaluation	154
6.9	Limitations	160
6.10	Summary and Conclusion	162
IV	Mobile Workspaces	163
7	PiraTop – Mobility for Hybrid Workspaces	165
7.1	Introduction	166
7.2	Related Work	167
7.3	Iterative Design Approach	176
7.4	PiraTop Overview	176
7.5	The Stationary Workspace	181
7.6	The Mobile Part	184
7.7	Implementation	191
7.8	Evaluation of the Visualization Styles	193
7.9	Evaluation of the Mobile Part	198
7.10	Design Implications	207
7.11	Summary and Conclusion	209

V	Conclusion and Outlook	211
8	OverTop – PeriTop Revisited	213
8.1	Introduction	213
8.2	The OverTop Concept	214
8.3	Interactions Enabled by OverTop	217
8.4	Summary and Conclusion	220
9	Summary and Conclusion	223
10	Future Research Directions	229
	List of Figures	233
	List of Tables	237
	Bibliography	239

Part I

Context and Preliminaries

Introduction

This chapter describes the context of this thesis and provides an overview of its structure, main contributions, and previous publications. It starts with a general motivation of the topic (section 1.1), provides an overview of the structure of the rest of the thesis (section 1.2) and concludes with an overview of the related publications (section 1.3).

1.1 Motivation and Context

The contributions in this thesis are situated in the context of knowledge work. Although there is no definitive definition of the term knowledge work [Pyöriä, 2005], it is generally associated with work that is mainly based on information analysis, the generation of ideas, problem-solving, and similar activities that heavily rely on the worker’s knowledge and the need for (external) information. As a result, a core aspect is the retrieval and management of information: From a high level perspective, the act of knowledge work can be described as retrieving the required information to fulfill the current task, executing the task, and then filing or distributing the results.

Historically, the primary medium for conducting this kind of work was paper and the main workplace of the knowledge worker was a desk in an office, often with additional storage space for information like shelves, filing cabinets, etc. Paper itself as a medium has some fundamental properties that explain its wide use that lasts even until today. First, it is cheap, lightweight, and nearly ubiquitously available. Second, it has some unique affordances that foster its use: It is flexible, can be folded and torn apart to form smaller pieces, and due to its low weight it is easy to carry around [O’Hara and Sellen, 1997; Sellen and Harper, 1997; Whittaker and Hirschberg, 2001]. Also, paper-based documents can be directly annotated with a pen, which is a very direct and lightweight way of working with a document. The desk then serves as a surface on which all paper documents can be laid out and flexibly structured.

1.1.1 Importance of Spatial Layout

The ability to spatially layout and structure documents on a desk has been shown to be a crucial affordance of paper that facilitates work processes [Kirsh, 1995]. Thus, one of the core goals of this thesis is to provide interaction techniques that allow the user to seamlessly employ concepts that rely on spatial organization, even when concurrently working with digital and physical content.

Based on the aforementioned unique properties of paper, over time, sophisticated organization techniques were developed to facilitate the management of large amounts of paper: For example, considerable research [Malone, 1983] has shown that piles are an important and frequently used means for structuring and managing workspaces. Typically, knowledge workers pile (and thereby group) sets of documents belonging to one thread of work – for instance, keeping track of the work progress by having “input” and “output” piles. One important aspect of the piling practice is the ability to order the documents within the pile (e.g., by priority). Besides piling, which is a rather simple, yet expressive, form of organization, there are also folders for grouping a set of documents more strictly, which in turn can easily be archived in a shelf.

Generally, this spatial way of organizing has been shown to be an essential aspect of the user’s productivity [Kidd, 1994; Kirsh, 1995]. Further research [Mandler et al., 1977] has shown that people are capable of coding spatial information into long-term memory in an “automatic” manner (e.g., one can remember an interesting newspaper article, knowing precisely that it was on page 3 on the lower right, without exactly remembering what it was about). Naveh-Benjamin [Naveh-Benjamin, 1987] found that this automatic coding may be limited to scenarios where the placement of objects happens deliberately, and users are willing to learn the structure. However, this can be assumed to be true for structuring one’s work environment.

1.1.2 Shift in Working Practices

The desk has been a place of innovation for a long time, as people have often tried to improve the work environment of knowledge workers. A fundamental idea to achieve this was and is automation. An early concept to automate knowledge work is the Memex system [Bush and Wang, 1945], depicted in figure 1.1, that dates back to the year 1945. It already provides concepts that are ubiquitous today, like linking between documents (cf. hyperlinks in the web

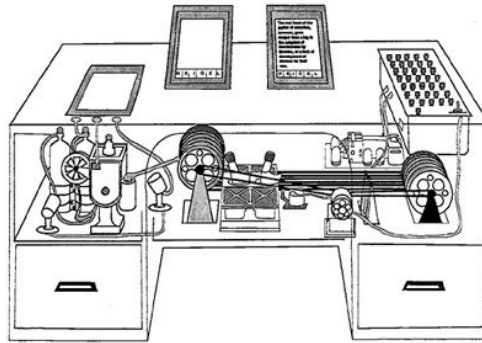


Figure 1.1: Vannevar Bush's Memex. (Source: http://resumbrae.com/ub/dms423_f08/06/memex.jpg)

or within documents), and convenient access to documents on small screens on the table (cf. windows and tabletop computers). Even the idea of remote operation was already present, as Bush states that *“It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works.”*

While the Memex was a visionary idea, the first implementation of an automated knowledge management came in a different form, namely the personal computer. With the advent of the personal computer, working practices changed: While before, there was mostly a single medium – paper – that was used to store information, information now exists as a digital or physical document (or even both). Digital documents are usually stored within the computer and are presented to the user on a vertical screen in front of the user, while physical ones are still on the desk as they used to be. A key factor for the success of personal computers was the transfer of well-known and established techniques from the physical world, using the desktop metaphor as the foundation for the user interface. Thus, concepts like the desk surface, folders, files, etc. are still present, albeit in a different form and with some limitations (e.g., most GUI interfaces do provide a kind of “piling” by allowing arbitrarily placing icons on the desktop or within folders instead of forcing them into a fixed grid layout).

Despite the effort to bring concepts from the physical world to the digital domain, a large gulf remains between the digital and physical worlds since both types of documents do not coexist within the same environment. As a result, it is, for instance, impossible to conveniently place a physical document next to an updated digital version and compare the changes. Also, since work practices now often comprise both types of documents, frequent context changes between

the digital and physical worlds are necessary. While it was proposed that the computer would lead to a paperless office, Sellen and Harper [1997] have shown that actually, the opposite is true: In their 1997 study, they found that a group of 25 workers (economists and administrative support people) at the International Monetary Fund spent 86% of their time working with paper documents, in contrast to 49% spent working with electronic documents (The excess 35% is time spent on working with digital and physical documents concurrently). Even though their study dates more than 20 years back, current figures (e.g., the CEPI ¹ lists about 72 million tons of paper produced in 1997 and about 78 for 2016) show that the consumption of paper has even increased since then, indicating that their findings are still valid today. The reason for this (still) heavy use of paper has been found to be the lack of flexibility of computer-based systems when it comes to annotating or flexibly arranging items [Luff and Heath, 1998; Luff et al., 1992; Steimle et al., 2010b].

Hybrid Knowledge Work

Knowledge work practices that comprise the concurrent use of digital and physical documents are referred to as *hybrid* knowledge work in this thesis. Consequently, tabletop setups that are subject to concurrent use of digital and physical objects are called hybrid tabletops or more generally hybrid workspaces.

1.1.3 Digital Tabletops and Tangible UIs

The development of interactive surfaces, especially tabletop computers, is intended to alleviate the aforementioned issues. Tabletops are computers that have a table-like form factor and have an interactive, touch-enabled, display embedded into the table surface (i.e., the whole table surface is a display). Thereby, they allow digital and physical documents to coexist again on the same surface, namely the desk: Digital documents are shown on the table surface display and can be moved around using touch gestures, and paper documents are placed on the display like they would be on a normal table. As a result, tabletops are becoming increasingly widespread, with researchers starting to deploy and study these systems in practice, for instance at homes [Kirk et al., 2012], classrooms [Kharrufa et al., 2013], or even biology labs [Tabard et al.,

¹http://www.cepi.org/system/files/public/documents/publications/statistics/2017/KeyStatistics2016_Final.pdf

2013]. It has been shown, for instance, that they foster student’s engagement in problem solving [Piper and Hollan, 2009]. Given the recent advances in creating high-resolution thin display technology and falling hardware costs, one can well anticipate that these systems will augment regular office desktops or meeting tables in our daily life and become an integrated part of the furniture.

Along a different line, tabletops are closely linked to tangible user interfaces. As a result, besides documents, tangibles, as a second class of important physical objects, can also be present on the tabletop and need to be integrated into the hybrid workspace. Thus, they need to be considered as well in the context of this thesis. Tangible user interfaces, which first appeared in the form of the Marble-Answering-Machine by Durrell Bishop in 1992 and were later studied more extensively by Ishii and Ullmer [1997], aim at removing the indirections of conventional user interfaces in which every digital object is manipulated through a mediator device like the mouse or keyboard. The use of such devices limits the expressiveness of the interface and the user’s capabilities to work with digital content. Tangible user interfaces provide direct physical access to functions and information by representing them via physical tokens (so-called *tangibles*). Since interaction does not need to be channeled through a mediator device, the user can employ the full richness of physical interaction, like bi-manual interaction, grabbing multiple objects at once, etc. The use and impact of tangibles has been studied extensively [Kirk et al., 2009] in general and in more specific cases, for instance for browsing large collections of photos [Terrenghi et al., 2008]. While generally tangibles have a positive influence, other studies have found a few drawbacks in tangible use, for example that interacting with content on the display via a tangible felt more indirect than touch input [Hancock et al., 2009]. However, indirect control via a tangible was found to be superior over non-tangible interaction for cases where an indirect interaction style is desired by users (e.g., users preferred to directly interact with a 3D object using touch for 2D translation, but preferred an indirect tangible trackball that was placed next to the object, for 3D rotation).

Despite providing a richer user experience, tangibles currently lack input and output facilities, i.e., they can be moved around to trigger input on the tabletop but their surface does not support touch input or visual output. Research has started to use, for instance, optical pipes [Willis et al., 2012] or integrated sensing [Sato et al., 2012; Schmitz et al., 2015] to mitigate this issue. However, these are complex to use and require modification of the tangible.

1.1.4 Tabletops in the Wild

By now, tabletops have been around for more than ten years on a conceptual and technological level, and even industrial manufacturers have started to produce and sell tabletops (e.g., SMART’s SMART Table 230i, Microsoft’s Surface Tabletop, Samsung’s SUR40). But still, we rarely see widespread practical use of tabletop computers in productive settings. Mostly, tabletops are used in showcases or as entertainment devices in hotel lobbies (e.g., in 2008, Sheraton Hotels deployed a few Surface tabletops in selected hotels) or a few very specialized applications like emergency planning [Chokshi et al., 2014; Doeweling et al., 2013; Kobayashi et al., 2006] or planning tasks in general [Bortolaso et al., 2013; Eden et al., 2002]. However, these applications are still in the research stage and not yet fully commercialized.

We argue that a large part of the problem hindering broader use of tabletops is the lack of integration of physical and digital objects, which the table-like form-factor suggests. The ideal tabletop would support all features of a regular table with the additional capability of showing digital content on its surface. Consequently, these devices must take two roles: First, an interactive display to visualize and manipulate virtual objects and second, a place to put everyday physical objects (e.g., books, printed documents, laptops, etc.). At a high level, this is exactly one of the conceptual strengths of tabletops and seemingly, they would thus enable the same richness of working practices that is known from purely physical settings in hybrid scenarios. However, in practice, three significant issues arise:

Occlusion by Physical Objects Digital and physical objects are fundamentally different despite coexisting in the same space. Since physical objects do have a dimension, the user can easily see them and even get a first impression of how many of them are there. For the digital ones, there is no correspondence – they are two-dimensional, and hence there is no immediate way of distinguishing a pile of 10 from a pile of 100. At the same time, they are still confined behind glass: The user can manipulate them via touch input but it is impossible to actually grasp them or lift them off the table surface. As a result, physical objects are dominant over the digital ones. This becomes especially obvious when placing a physical object on top of a digital - the digital object is immediately occluded, and there is no way of knowing it is there. Further, as the number of physical objects increases, the remaining digital space becomes limited.

Lack of Stacking Support Since digital objects can only exist in the display layer, more advanced organization techniques like piling are severely limited: Digital objects have to be always separate from physical objects (digital ones are “inside” the table, physical ones on top). While acceptable for an unordered pile, actions like ordering a mixed set of documents alphabetically become impossible.

Competition for Space Objects present on the table play different roles in the context of the ecosystem and compete for space: As mentioned, physical objects occlude the display, making it difficult to keep important digital content salient. At the same time, there is a difference between the physical objects themselves, e.g., between (i) tangibles that represent digital actions, (ii) physical documents or other currently task-relevant objects, and (iii) currently irrelevant objects (e.g., plants, coffee mugs, or a printer placed on the table, etc.). Current tabletop implementations tend to assume a clean surface on which only relevant objects are present. “Relevant” in this context is usually limited to tangible objects only, as they have a meaning in the context of the interface. Even objects from category (ii) that might be relevant to the user usually cause problems (e.g., false touch events)

1.1.5 Mobile Work

In addition to the evolution of stationary workplaces, mobile work at different locations has become increasingly popular and widely accepted by employers: People are no longer working only at stationary offices with fixed personal workplaces but also from their home offices or while traveling (e.g., while traveling in a train or from a customer site). This has been not to the least enabled by the technological advances in producing highly mobile devices like smart-phones, laptops, etc. that provide ubiquitous access to information. Further, mobile work has been shown to have several advantages, like increased employee satisfaction [Gajendran and Harrison, 2007], better work-life-balance [Hill et al., 2003], and lower cost for employers [Ruth and Chaudhry, 2008].

Remote Access However, working remotely poses an additional set of challenges, like setting up an appropriate environment with access to corporate resources, which has led to a slower adoption rate [Ruth and Chaudhry, 2008]. Especially in the context of the aforementioned hybrid work environments, setting up the work environment becomes particularly cumbersome: While in a

paperless office, digital content could be made easily accessible from anywhere, the physical paper-based part is not that flexible. This becomes obvious when taking into account that a worker might not only need a small and specific selection of documents (like for a meeting) that is still easy to carry but might rather need access to a large number of documents. At that scale, it would be cumbersome to transport paper documents and doing so would also destroy the spatial layout since the user would have to pack everything into a briefcase for transport. To overcome these problems, a way of *remotely accessing* the whole stationary hybrid work environment from remote locations would be desirable (cf. remote desktop software for computers). Thereby, the spatial layout remains intact and the need to decide upfront which documents might be required for a meeting would be eliminated.

1.1.6 Scope of this Thesis

This thesis aims to integrate physical and digital information more closely and seamlessly, thereby tackling the challenges outlined before (occlusion, lack of stacking support, competition for space, and remote access). To provide a consistent setting, all contributions are placed within the context of a knowledge work or office ecosystem. The first set of contributions presented in parts I to III focuses on the use of tabletops as stationary hybrid office desks. The last contribution presented in part IV focuses on the mobility aspects by providing a spatial layout-preserving remote access approach, thus expanding the setting from the stationary office to off-site locations.

Generally, the contributed concepts are technically not limited to the office desk environment described and used within this thesis but could also be extended to additionally support other aspects of an office (e.g., shelves, file cabinets, etc.). Going one step further, it is possible to apply the presented concepts to other domains as well, for instance, in the context of digitally augmented hybrid lab benches or assembly workplaces. Of course, this would require the presented concepts to be modified and adapted to fit the respective scenarios, which is, however, beyond the scope of this thesis. As a result, throughout the rest of the thesis, the more specific term “documents” is used in most cases instead of the more general term “objects” to be consistent with the knowledge work setting. This does, however, not imply a general limitation of applicability to documents only.

1.2 Thesis Structure

After this introduction, the thesis starts with an overview of the stationary hybrid workspace setup (**chapter 2**) that is used throughout the thesis and establishes required terminology as well as a design space that the next two parts of this thesis are located within.

Subsequently, the challenges outlined in the Motivation section are addressed by contributions in three different areas that also form the three main parts of this thesis: Stationary Hybrid Workspaces, Layout for Hybrid Environments, and Mobile Workspaces.

Stationary Hybrid Knowledge Work In this first part, we first investigate the problem of occlusion, since it is one of the fundamental issues when concurrently using digital and physical documents. In **chapter 3**, we propose a set of interaction and visualization techniques that build on the concept of peripheral displays which leverages not only the desk surface but also the surface of the occluding object to provide awareness of and access to occluded objects. We further report the results of an evaluation comparing our new techniques with a state-of-the-art tabletop-only system. In **chapter 4**, we present a concept to further interweave digital and physical documents by providing support for hybrid stacking (i.e., digital and physical documents can be arbitrarily placed on top of each other). Again, we present the interaction and visualization techniques followed by an evaluation.

Layout for Hybrid Environments In the second part, we tackle the problem of automatically laying out digital content based on the environment. Since the surfaces of objects are unknown at the time of designing a system, an approach is needed to identify areas suitable for projection dynamically. In **chapter 5**, we propose a measure to assess projection quality that flexibly combines multiple factors, thereby allowing it to be applied in various domains. We conclude the chapter with a brief technical evaluation assessing its real-time capabilities. Building on the projection quality measure, in **chapter 6**, we present an integrated framework that provides dynamic user interfaces on and around tangibles that react to surrounding objects by adjusting the layout of displayed content to avoid occlusion and interference. The final system was evaluated technically and in a first user study.

AR Remote Access	AR above the Table	Chapter 7 8	Part IV V
	Integration of Tangibles and Automatic Layout	Chapter 6	
	Free-Spot finding for Hybrid Environments	Chapter 5	Part III
	Hybrid Stacking	Chapter 4	
	In-Situ Occlusion Resolution	Chapter 3	Part II
	Peripheral Displays & Hybrid Tabletop Environment	Chapter 2	Part I

Figure 1.2: Thesis structure, contributions, and their relations.

Mobile Workspaces As mentioned in the introduction, mobile work has become increasingly popular. In the third part, we present a concept for augmented-reality-based remote access to hybrid work environments in **chapter 7**. Augmented reality makes it possible to maintain the spatial layout during mobile work and provide fully three-dimensional access. The presented system was evaluated in two user studies. The first study compared the different visualization techniques developed; the second evaluated the system as a whole.

Finally, in **chapter 8**, the initial hybrid work environment concept is revisited by considering additional possibilities provided by the use of augmented reality glasses at the stationary workspace. The thesis concludes with a summary and outlook. Figure 1.2 summarizes the core contributions outlined above.

1.3 Publications

Parts of the content presented in this thesis have been previously published in the proceedings of international peer-reviewed conferences, such as the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), and journals like the Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT). As a result, content has been used verbatim from the respective publications. The chapters and the corresponding publications are listed in table 1.1.

Contribution statement For all publications that are listed in table 1.1 and that form the basis of this thesis, the author of this thesis initiated and led the project and developed the ideas and concepts presented. FlowPut was a joint project together with Martin Schmitz, who provided the use case for 3D-printed objects and was therefore also involved in developing the details of

Chapter	Publication(s)	Type
2 – PeriTop	Riemann et al. [2013]	EA
3 – ProjecTop	Riemann et al. [2015b]	FP
4 – StackTop	Riemann et al. [2015a]	EA
	Riemann et al. [2017]	WP
5 – FreeTop	Riemann et al. [2016]	EA
6 – FlowPut	Riemann et al. [2018b]	J
7 – PiraTop	Under Submission	
8 – OverTop	Riemann et al. [2018a]	EA

Table 1.1: Overview of publications. The publication type is abbreviated as follows: EA – Extended Abstract, WP – Workshop Paper, FP – Full Paper, and J – Journal

the concepts for the optimization-based layout that go beyond what is already presented in FreeTop. The coauthors of the publications also contributed to the conceptual design of the systems and writing of the papers. The implementation of the concepts has been supported by bachelor and master students whose theses were supervised by the author of this thesis: Mathias Mettel for the improved touch detection presented in PeriTop, Christoph Bauer for ProjecTop and StackTop, Alexander Hendrich for FreeTop and FlowPut, and Christoph Niese for PiraTop.

Other publications Besides the contributions listed above that are the foundation of this thesis, the author of this thesis has also coauthored several other publications in the field of occlusion awareness [Khalilbeigi et al., 2013], hybrid tabletops [Mühlhäuser et al., 2014], augmented reality enhanced tangible interaction [Günther et al., 2018], interaction with geospatial data on tabletops [Döweling et al., 2016], tangible-based interaction for large displays [Günther et al., 2017], resizable displays [Khalilbeigi et al., 2010], and 3D-printing [Schmitz et al., 2016]. Since some of the publications in the field of tabletops are partially relevant to this thesis as well, they are cited and discussed in the respective related work discussions.

PeriTop – Hybrid Tabletop Environment

This chapter introduces a concept for *peripheral displays* for hybrid tabletop environments named “PeriTop”. The chapter sets the general framework for all contributions made in the following chapters and explains important terminology and concepts used throughout the rest of the thesis.

The chapter starts with an introduction (section 2.1) and describes relevant related research in section 2.2. Section 2.3 establishes relevant terminology and general technical preliminaries before the details of the PeriTop concept are described in section 2.4. Based on the presented concepts, a set of use case scenarios is developed in section 2.5 of which three are explored more deeply in the following chapters of this thesis. The chapter concludes with the discussion of a few selected implementation issues in section 2.6 and a summary in section 2.7.

2.1 Introduction

In the previous chapter, we argued that the use of digital tabletops as a general purpose working desk is subject to a range of problems. Especially when used in combination with physical objects, issues like occlusion, inability to arbitrarily stack objects, competition for screen space, and a lack of in- and output for tangibles occur.

The presence of physical objects on tabletops thus poses several challenges. The most notable is the considerable decrease of screen-space available for interaction with digital objects. Not only does the footprint of physical objects directly obstruct the display, but their casual arrangement on the surface makes the still visible display areas harder to use. This is because the remaining free display areas are usually irregularly shaped and do not provide large rectangular free spaces. This becomes even worse for tall physical objects

that shadow even more display area from the user’s perspective. As a result, physical objects may partially or entirely occlude digital objects, resulting in a loss of awareness of the occluded objects.

Work has emerged to deal with these issues, e.g., by detecting the footprint of physical objects [Freeman and Brewster, 2013] and avoiding occlusion [Cotting and Gross, 2006]. Other researchers have addressed the problem by providing support for staying aware of and accessing digital objects [Furumi et al., 2012; Javed et al., 2011; Khalilbeigi et al., 2013]. The typical approach is that the occluded objects are transformed into icons or other content-wise reduced forms and visualized next to the occluding object. While this pioneering and motivating work established first design principles to mitigate the problem of physical occlusion in hybrid settings, it is not clear how these approaches extend to more cluttered workspaces where many physical objects consume considerable space on the tabletop display. In such cases, the already scarce display space is further reduced by the visualizations used for occlusion mitigation. Thus, the tabletop display, which is the user’s primary interaction surface, might even become unusable for productive work. Adding to this, occlusion mitigation techniques that rely on displaying content at other than the actual locations, alter the perceived spatial layout of the desk, which might lead to further usability problems.

In this chapter, we extend the tabletop-based interaction and display space to the surfaces of physical objects present on the tabletop by additional top-projection facilities on a back-projected tabletop system. The tabletop provides the primary high-resolution interaction surface for the user to work on while the top-projection provides a secondary class of “peripheral” displays on the surfaces of physical objects. These peripheral displays can then be used to solve the problems mentioned above. By using the surface of an object as secondary display, we do not need to consume additional space on the tabletop screen. Further, using surfaces of objects as displays allows us to provide visual output and touch input on tangibles seamlessly.

2.2 Related Work

The work presented in this chapter is related to the fields of augmented desktops and spatial organization, which we discuss in the following.

2.2.1 Augmented Desktops

Working with documents on a computer has revolutionized knowledge work and work-practices in general. As a result, there is a shift towards the use of digital document management techniques. However, as discussed in chapter 1, paper still plays an important role, and, as a result, tasks today often still require printed documents and will probably do so for the foreseeable future [Sellen and Harper, 2003]. Thus, most work tasks can be considered as hybrid with respect to the documents being used, as they employ physical and digital documents concurrently. However, when using a desktop computer with a vertical screen, there is no integration between the physical and digital worlds. Thus, augmenting the physical desktop has a long tradition in HCI. As such, a lot of research exists that explores the digital augmentation of either (single) documents or the table as a whole.

Document-Level Augmentation

Research has emerged to solve the integration issues by means of projection. Using projection systems allows digital and physical documents to coexist by projecting digital documents (or more generally content) into the physical world. Many efforts have been made to create *augmented paperwork* with digital projection or input, thereby allowing the user to interactively add digital content (e.g., annotations or animations) onto or next to physical documents: The DigitalDesk [Wellner, 1993], PenLight [Song et al., 2009], FACT [Liao et al., 2010], and Paper-Top [Mitsuhara et al., 2010] are systems that project onto printed documents, which allows, for example, the user to interactively add digital content [Liao et al., 2010; Mitsuhara et al., 2010; Song et al., 2009; Wellner, 1993], show animations [Mitsuhara et al., 2010], or do highlighting [Liao et al., 2010]. The EnhancedDesk [Koike et al., 2001] uses a projector to display related digital content (e.g., interactive simulations) next to physical documents. Since using a projector is often cumbersome (i.e., it needs a tracking system for objects, and the projector needs to be calibrated so that the projection matches the position of the object), research has also explored the direct integration of electronics into paper in order to augment physical documents. However, this is currently limited to highlighting predefined areas [Klamka and Dachsel, 2017].

While inspiring, these systems aim at augmenting individual paper documents with additional digital facilities and do not integrate “self-contained” (i.e.,

documents that have meaning on their own and are not only digital add-ons to physical documents) digital and physical documents. However, to be used as a general purpose augmented desktop or in the context of tabletops, it is necessary to provide support for self-contained digital objects as well. Further, these approaches are limited to augmenting a single physical document at a time, but do not take into account concurrent presence of multiple documents that is common on an office desk. Also, most of the presented systems require a pen or other special input device and do not support direct touch interaction with the projected content.

Interactive Desks and Beyond

Going one step further and not only augmenting at a document level, researchers have explored the use of interactive tables or surfaces in general, where the whole table surface is a display and can also host self-contained digital objects. Pioneering in this field is the *Office of the Future* by Raskar et al. [1998], an office augmented with digital projection everywhere, allowing digital information to be put on any physical object or surface. Yet, the Office of the Future only supports working with digital documents and does not consider physical documents or the desk as specific working environment.

Rekimoto and Saitoh [1999] developed a system that allows digital objects to “leave” devices like laptops by means of top-projection on desk and wall surfaces. Using their system, drag-and-drop of content out of the computer, onto tables, walls, etc. is possible, thus making the workspace spatially continuous (hyperdragging) and breaking the boundary of the computer screen for digital objects. Again, this work focuses on digital documents and does not take into account physical documents (although the system can link digital artifacts to physical objects).

Another example that focuses more on a single surface is the Pictionaire system [Hartmann et al., 2010], which is a projected tabletop system for integrating digital and physical media. Projection on physical objects is used by Pictionaire to facilitate tracing of digital content into sketchbooks by projecting the original digital content on the book. However, there is a specific focus on drawing and hence general knowledge work (e.g., with the requirement of textual documents or digital annotations on documents) is not well supported.

In a similar direction, Kim et al. [2004] bridged the gap by allowing printed

physical representations of documents to be linked to their digital counterparts. This allows (digitally) searching for physical objects on the desk or remotely accessing the digital model of the physical desk. To do so, it uses computer vision-based tracking to keep track of the changes in alignments and physical stacks. While the concept is aimed directly at desks and multiple documents, there is no direct visual feedback on the desk and no support for digital documents at all.

Besides the use in office environments and document management, interactive surfaces have also been explored in the field of lab benches, e.g., in the form of the eLabBench [Tabard et al., 2012]. The eLabBench supports working in a biology lab by providing augmentations for test-tube racks, digital note-taking capabilities, access to information, and a direct photo capture button to capture a “screenshot” of the physical setting.

Along another line, there are also some systems, like PaperLens and spatially aware tangible displays [Spindler, 2012; Spindler et al., 2009, 2012], that use top-projection and passive devices to extend the interaction space from the surface of the tabletop to the space above the table. This allows, for example, naturally browsing three-dimensional information spaces.

Interactive desk systems usually use a single means of display, mostly realized using a projector. While this allows for easy deployment, projection usually has some drawbacks like sensitivity to daylight and occlusion by objects or even the user’s own body.

Requirements and Summary

In the following, we present a set of requirements for providing a seamless integration of physical and digital documents in hybrid environments derived from reviewing the related work and use them to compare the related work (see table 2.1).

R1 Multiple physical documents Knowledge work requires retrieving information from multiple sources and as a result, multiple documents are concurrently placed in the working environment. Thus, the concurrent tracking and augmentation of multiple physical documents must be supported.

R2 Self-contained digital documents To provide a deep integration between digital and physical documents, self-contained digital documents

	Multiple Physical Documents	Self-Contained Digital Documents	Digital Annotations	Spatial Layout	(Multi-)Touch Input
<i>Augmented Documents</i>					
Wellner [1993]	○	◐	◐	○	●
Song et al. [2009]	○	○	●	○	◐
Klamka and Dachsel [2017]	◐	○	●	○	◐
Liao et al. [2010]	○	○	○	○	◐
Mitsuhara et al. [2010]	○	○	◐	○	○
<i>Interactive Desks</i>					
Raskar et al. [1998]	◐	●	○	●	○
Hartmann et al. [2010]	●	●	◐	●	◐
Rekimoto and Saitoh [1999]	◐	◐	●	●	●
Kim et al. [2004]	●	○	○	●	○
Tabard et al. [2012]	◐	●	○	●	◐
<i>Spatially Aware Displays</i>					
Spindler et al. [2009]	○	●	○	●	○
Spindler [2012]	○	●	○	●	○

Table 2.1: Properties of selected augmented desktop systems. ● indicates that a requirement is fulfilled, ◐ indicates partial fulfillment, and ○ that a requirement is not fulfilled.

(e.g., videos, text documents, etc.) need to be supported as well.

R3 Digital annotations Digitally annotating a physical document using, for instance, free-hand annotations or by linking digital content to it should be supported.

R4 Spatial layout The system needs to be aware of and account for the spatial layout of the documents present.

R5 (Multi-)touch input Multi-Touch input without the need for additional devices like styluses or digital pens is one of the most important input modalities in the context of tabletops. As a result, it should be supported on the tabletop surface as well as on the surfaces of physical objects.

In summary, the presented systems are well suited for augmenting physical documents with digital annotations or providing digital documents on physical tables. However, they only coarsely cover the concurrent use of self-contained digital and physical documents that is addressed in this thesis.

2.2.2 Spatial Organization

As mentioned in the introduction of this thesis, sophisticated organization strategies have evolved over the past centuries to efficiently work with physical documents on desks. Hence, the use of desks to structure and organize information in the analog world has been extensively studied. Also, research has focused on the use of spatial organization in the digital world.

Spatial Organization in the Physical World

Malone [1983] identified two basic strategies for working with documents: Piling and filing. Filing refers to the structured sorting of documents into folders or cabinets, and thereby removing documents from the active workspace and storing them. Piling, however, is an essential means of structuring documents on workplaces in a very flexible way. Piles not only allow users to group documents belonging together, but they also enable classifying them by having piles for different types of documents (e.g., documents to be worked on in one pile and documents already processed in another). Since piles are not as tightly coupled as – for example – binders, it is easy to relocate individual documents to different piles naturally and directly. Depending on how tidy the user wants his organization structure to be, it is even possible to casually place documents across two piles to denote that they belong to two categories at the same time. Piles also serve as a means of spatial organization, i.e., the spatial location of a pile or a document within a pile can have a specific meaning to the user (e.g., documents at the bottom of a pile are older than those at the top). Mander et al. [1992] confirmed the importance of piling to manage information quickly and informally in a spatial manner. They note that this is especially the case for information that is not strictly hierarchical.

As mentioned, piles not only serve as a means of grouping, but they also have a spatial component (i.e., their precise location on the desk). Research has shown that this spatial position is an important aspect of the usefulness of piles. Kirsh [1995, 2001] have studied the use of space and concluded that the spatial layout could provide entry points to start or resume tasks, provide an

action landscape, and serve as a coordination mechanism. This reduces the user’s memory load by externalizing. Kirsh distinguishes two types of people in this context, namely “neats” and “scruffies”. They differ in the structure of entry points and canonical locations of objects: “Neats” prefer a tidy desk with fixed locations for objects. In contrast, “Scruffies” use a chaotic desk layout. For scruffies, messy desks hold much information about activities, as they feel that strict and simple organization schemes do not fit the dynamics of their activities.

More recently, Bondarenko and Janssen [2005] have shown, in a two-year ethnographical study that considered physical as well as digital documents, that the ability to manage and group documents is strongly related to the management of tasks and that digital tools often do not support this grouping adequately.

Further research [Mandler et al., 1977] has shown that people are capable of coding spatial information into their long-term memories in an “automatic” manner (e.g., one can remember an interesting newspaper article, knowing exactly that it was on page 3 on the lower right without exactly remembering what it was about). Naveh-Benjamin [1987] found that this automatic coding may be limited to scenarios where the placement of objects happens deliberately, and users are willing to learn the structure. However, this can be assumed to be valid for structuring one’s work environment.

Spatial Organization in the Digital World

Researchers have explored spatial organization in the digital world as well. Patten and Ishii [2000] conducted a study of spatial organization in graphical and tangible UIs focusing on spatial memory. To help memorization, people employed spatial arrangement strategies that were used in tangible UIs, but not (that much) in purely graphical interfaces. For instance, grouping by placing objects spatially close, ordering objects along an “axis”, or “reference-based positioning” in which the relative or absolute placement of objects codes information (e.g., geographical aspects by placing left for west, etc.) were more likely to be used in the tangible context than in a graphical user interface. They concluded that the haptic dimension of tangibles fostered adapting organization schemes that were found difficult to use in GUI interfaces. Further, it has been shown that participants were better at recalling the location of objects in the tangible condition than in the graphical interface. It was mentioned that in

addition to the spatial memory, motor memory is advantageous for physical items. Further, when placing physical items, one must take context or other objects into account (e.g., a large document cannot be placed over a coffee mug in the real world). This leads to a more conscious and deliberate placement of objects, resulting in better memory [Naveh-Benjamin, 1987]. Also, a physical desk allows placing objects at several locations relative to the user (e.g., left, right, etc.), while on a computer screen, they are always on the screen in front of the user.

The concept of spatial memory for document organization has been employed in three notable systems that are described in the following.

The DataMountain concept [Robertson et al., 1998] uses spatial memory for document management. Robertson et al. show that spatial memory can be successfully leveraged for document management, outlining that spatial memory often aids in finding things in the real world: “*For example, when we place a piece of paper on a pile in our office, we are likely to remember approximately where that paper is for a long time*”. One key to this efficiency is that the documents can be arranged in an informal, personal way: “*The user study also suggests that spatial memory does, in fact, play a role in 3D virtual environments. We often heard subjects say things like ‘it’s right here’, or ‘I know it’s back there’, and move directly to the location of the page.*” This led to reduced times needed for storage and retrieval. Further, the error rate when searching and retrieving a document lowered because of the influence of spatial memory, according to Robertson et al.

The key role of spatial layouts and spatial memory has even led to desktop systems being built based on this concept: The Bump-Top desktop system [Agarawala and Balakrishnan, 2006] presents the user with a 3D desktop with piling instead of filing as the fundamental organization structure. To be more realistic, it supports physics-based interaction (i.e., the documents have a mass and fall like real-world objects if dropped from a height). The system also does not force items on the desktop in a specific layout without user interaction, unlike the “snap-to-grid” behavior of many desktop systems today. This is because Agarawala et al. consider spatial orientation and position to be “usually meaningful”. Piles are for instance automatically ordered chronologically by putting new documents on top – A behavior that is prevalent in the real world but missing in today’s desktops.

To bridge the gap between the digital and physical worlds, Kim et al. [2004] developed a system that tracks the layout of a workspace and allows remotely

accessing the digital version of a document on the desk. Since the spatial layout is tracked and preserved, the user can access his desk via a spatially correct representation of his physical desk shown in a desktop application. The system, however, only works in one direction – from the physical desk to the digital representation – and does not allow incorporating purely digital documents into the layout.

In this chapter, we set the foundation to expand the possibility of creating complex spatial layouts from purely digital or purely physical environments to hybrid usage scenarios on tabletops. Thereby, the user is enabled to benefit from the advantages spatiality provides for productivity in a wide range of knowledge work scenarios.

Requirements and Summary

In the physical world, spatial layout has been shown to be a crucial factor for the productivity of knowledge workers. The spatial layout of documents plays a key role in many common organization techniques like piling, filing, or otherwise arranging documents on the workspace. Spatial layout is used, for instance, to group documents, manage different threads of work, or keep track of the workload. Further research has shown that humans automatically encode the layout in their long-term memory when they consciously place an object.

In the digital world, support for comparable techniques is still very limited. In the following, we present a set of requirements for hybrid systems supporting spatial organization in hybrid environments derived from reviewing the related work and use them to compare the related work (see table 2.2). Since they focus on the digital domain, the related systems all lack support for combined use of digital and physical documents.

- R1 **Digital and physical documents** Knowledge workers have to consider digital and physical information sources, often in combination. The concurrent use of both digital and physical documents must be supported
- R2 **Complex organization** In the context of purely physical documents, complex organization techniques like piling or stacking exist. They need to be supported in hybrid environments as well to form a continuous workspace.

	Digital and Physical Documents	Complex Organization	Non-Document Objects
Robertson et al. [1998]	○	●	○
Agarawala and Balakrishnan [2006]	○	●	○
Kim et al. [2004]	○	●	○

Table 2.2: Properties of selected systems that focus on spatial layout. ● indicates that a requirement is fulfilled, ◐ indicates partial fulfillment, and ○ that a requirement is not fulfilled.

R3 Non-document objects Work environments do not only consist of documents, but other objects as well (e.g., coffe mugs, office equipment). Systems should be aware of them and react sensibly.

2.3 Preliminaries

The concept of using peripheral displays together with a primary display was motivated and influenced by a set of factors, like the general use of workspaces, the different roles of objects, and issues caused by visual interference and occlusion, as well as strategies that exist to circumvent them, that we examine in the following.

2.3.1 Workspace Model

Research has shown that the surface of a desk is not used equally as a whole by users, but rather in different zones with different purposes (see figure 2.1). The area close to the user’s position is the main working area where the user interacts with objects, which is followed by an intermediate area. The largest part of a desk (depending on its size) is used as a storage area, where users pile away currently unneeded documents or place devices like monitors, printers, etc. [Toney and Thomas, 2006].

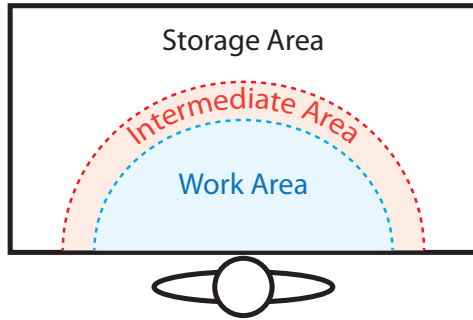


Figure 2.1: The different areas on a working desk.

As a result, when thinking of digital tabletop use in general office scenarios, these different zones imply different requirements for input and output capabilities: Although a high-quality high-resolution display with precise touch or pen input is inevitable in the working area, in the storage zones a lower display quality might be sufficient while still enabling the user to recognize or skim documents. At the

same time, shadowing by the user is mostly caused in the working area where the user moves above the table, while in the storage area, interactions only rarely take place (e.g., when the user grabs a new document).

2.3.2 Role of Objects

The objects within the domain of a hybrid tabletop environment have different properties and roles. At a high level, they split into physical and digital objects. However, within these two groups, there are further subclasses of objects. Table 2.3 summarizes the properties of the different classes.

Physical Objects

Physical objects can be categorized into three categories: tangibles, documents, and other objects.

Tangibles Since tangibles are physical representations of digital functions or information that is present in the digital workspace, they have an inherent meaning in the context of the tabletop user interface. Because they are meant for easy manual manipulation, they are usually rather small and can be moved easily. Tangibles are usually only relevant in the context of a specific application, so in the context of other applications, they become “other” objects.

Documents This category includes all objects that carry or represent self-contained information relevant to the user (text, photos, drawings, etc.). Since, in the domain of knowledge work, the majority of these objects are usually (text-)documents, the category is named documents. Unlike tangibles, these

	Size	Mobility	Life-span
<i>Physical Objects</i>			
Tangibles	small	high	short
Documents	small-medium	medium	medium-long
Others	small-large	low	short-long
<i>Digital Objects</i>			
User Interface (UI)	small-medium	low	short
Documents	small-medium	medium	medium-long

Table 2.3: Summary of important properties of the different object classes.

objects do not have a meaning in the context of the user interface on the tabletop, but are meaningful for the user. Since documents are the main working objects for a knowledge worker, they form the largest part of physical objects. Compared to tangibles, documents are therefore covering more screen space and are not as mobile as tangibles (e.g., a big pile of documents is harder to move than a small plastic tangible).

Other Objects This category comprises all other objects that might be found on a table that are neither tangibles nor documents – for example, plants, a printer, or a monitor. These objects don’t carry any information on their own but are nevertheless important artifacts for work practices. The objects in this category are often quite large and heavy (e.g., printer) and are hence mostly static objects that are seldom, if ever, moved. Despite often having a large footprint, the number of such objects usually is low compared to the number of document objects.

Digital Objects

For digital objects, there are two main categories, namely user interface and documents.

User Interface User interfaces are considered to be all digital objects that control an application or are tightly linked to its functionality, for instance, interactive widgets displayed around a tangible, tool palettes, etc. They are usually only present in the context of a specific application, just like tangibles in the physical world. However, unlike tangibles, they automatically vanish if the application is closed.

Documents Just like in the physical world, this category contains all self-contained (i.e., that have a meaning on their own, unlike – for instance – digital annotations that are only meaningful in the context of the object they annotate) digital content objects that are present on the workspace like text documents, images, videos, e-mails, etc. In an integrated workspace, they should not vanish upon closing – for instance – the text editor, but remain on the surface to be able to be sorted into stacks or otherwise arranged.

2.3.3 Occlusion

Occlusion has been found to be a cause of problems (e.g., losing awareness of occluded objects, cumbersome access, etc.) in hybrid tabletop environments if there is no system support [Steimle et al., 2010a,b]. In the following, we use the term “*occluder*” for an object that *occludes* another object and “*occludee*” for an object that *is occluded* by another object. Generally, there are different types of occlusion and from an object-type perspective, we can distinguish three types: a digital object occluding another digital object (digital/digital), a physical object occluding another physical object (physical/physical), and a physical object occluding a digital object (physical/digital).

The primary focus of this thesis is on physical/digital occlusion, since it is the most challenging type of occlusion: For digital/digital occlusion, all involved objects can be controlled by software, and concepts like, for instance, the Exposé mode that is well-known from desktop computers, can be used to provide an overview and awareness of occluded objects. In the case of physical/physical occlusion, the physical properties of the objects provide a minimum awareness of their presence, e.g., due to its height, a pile of documents can be easily perceived as a pile that contains multiple documents. However, in the case of physical/digital occlusion, the physical occluder can completely cover the digital content, leaving no awareness of the occluded digital objects. Despite the primary focus on physical/digital occlusion, some of the techniques presented in this thesis can be and are applied to the other types as well.

When looking at occlusion from the digital perspective, it is clear that user interface elements should never be occluded, since this category controls the currently running application. Hence, digital objects in this category are very important and always need to be accessible by the user, either directly by showing the full set of controls or through a proxy, e.g., by showing a smaller icon that toggles the visibility of the represented UI part. For documents, the

situation is much more differentiated: Since documents are long-living objects, the user needs to be able to occlude them; otherwise organization strategies like piling would be impossible. In this case, the occlusion does not cause any major issues and is on-purpose. At the same time, access to and awareness of digital objects is massively hindered when they are occluded.

From the physical perspective, the impact of occlusion also differs between the categories: Even though tangibles cover some screen space, this usually does not cause any problems, as they are tightly integrated into the digital workspace and the user interface reacts to them in a meaningful way so that the user can conveniently use them. Also, as mentioned, they are highly mobile and can be relocated easily. While there might be restrictions imposed by the tabletop system regarding their position (e.g., a rotary control to control a digital object must be placed on the object it controls), in such cases, the system is usually designed in a way that takes into account this type of occlusion. Hence, occlusion caused by tangibles can be considered acceptable. Physical documents are quite similar to digital documents – they are present for a longer time and might meaningfully occlude digital documents. However, they might also be thoughtlessly placed on the table without any intention, thereby occluding unrelated digital content, which in turn might be harmful if the user loses awareness. For the “other objects”, the severity must be seen relative to their dynamics: Fixed objects like a printer also occlude the surface, but since they are always present and fixed in one position, one could consider the area covered as a non-display area, just as if the display was shaped to leave out that areas. For objects that are more dynamic, like a casually placed laptop, the considerations for documents apply.

In summary, we can say that in the context of documents, occlusion is at least sometimes meaningful (to group or stack things) and not a bad thing that should be avoided. Yet, in the physical world, there is always at least some kind of awareness of occluded objects in the case of not-meaningful occlusion that is missing for digital documents (cf. their lack of dimension). Table 2.4 summarizes the most important properties of physical/digital occlusion.

2.3.4 Interference

While occlusion (in the physical/digital context) is the result of physical objects being placed on top of digital content, interference is the effect of digital content being displayed on top of a physical object. The term interference is used since

Occluder	Duration	Impact
Tangibles	During application use	Usually acceptable
Documents	Medium to long term	The user loses awareness of occluded objects. However, might be useful for grouping (e.g., piling).
Other (small/mobile)	Short to long term	The user loses awareness of occluded objects.
Other (large/stationary)	Long term	The display area can be considered non-existent (e.g., under a printer).

Table 2.4: Summary of properties of physical/digital occlusion.

the content of a physical content is not completely hidden by the projection but rather the projection and the texture of the physical objects interfere visually. Since the surfaces of arbitrary objects are used as a peripheral display by using projection, either an unstructured, light-colored object-surface is needed, or the projected content should be only light and unstructured (e.g., uniformly colored areas for highlighting). Otherwise, the projected content interferes with the texture on the surface of an object. While this does not pose a risk of losing awareness as both the projection and the content of the object remain visible, it renders both unusable (e.g., in case of text projected onto a printed text, both become hard if not impossible to read).

2.3.5 Dealing with Occlusion and Interference

To avoid the issues caused by occlusion and interference, the tabletop needs to react to them. Strategies for system reactions can be categorized into three areas:

Indifference This is the most trivial strategy in which the tabletop simply ignores the occlusion or interference and does nothing. The reasoning behind this strategy is that if an object is occluded, the user must have done it, and then it was on purpose. While this might be the case and likely does not cause any issues for short-term use, over more extended periods, the user might not remember what and where is occluded, leading to usability issues. However, if the system could infer whether an occlusion happened intentionally or not, it might be a viable reaction at least for short-term occlusions to avoid additional

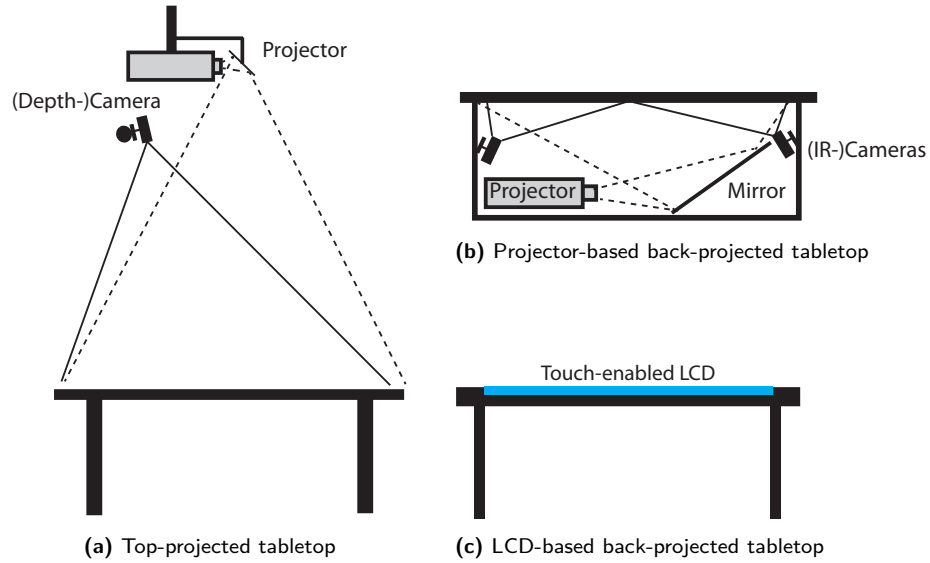


Figure 2.2: Schematic overview of the main types of tabletop implementations.

clutter caused by the other strategies.

Avoidance This strategy aims at avoiding occlusion or interference by relocating digital content that is about to be occluded to a free spot on the table so that it remains visible and accessible for the user. Given the considerations on the role of objects, this is a suitable strategy for user interface objects. However, when considering documents, this strategy would make piling impossible. Also, when considering a more substantial number of documents, there might not even be an empty free spot to move the document to. Hence, it is only suitable for a small, selected number of objects at the same time.

Awareness Strategies in this category do not actively avoid occlusion or interference. However, they aim at supporting the user in such situations by providing means of access to and awareness of the digital objects. As a result, the user still can pile documents but is reminded of the now invisible objects.

2.3.6 Technical Preliminaries

There are three fundamental ways to implement digital tabletops: Top-projection, back-projection and using LC-Displays. Since, for the work presented here, there is no substantial difference between actual projector-based back-projection, and using an LCD, both are subsumed under back-projection

	Top	Back
Immune to Environmental Light	●	●
Immune to Occlusion by Objects	●	○
Immune to Occlusion by Body	○	●
Integrated Device	○	●
High-Resolution	●	●
Covers Large Areas	●	●

Table 2.5: Summary of important properties of top- and back-projection-based tabletops. ● indicates that a property is given, ● indicates partial presence, and ○ no presence.

in the following. While both top- and back-projection lead to displaying content on a surface, they have different properties, advantages, and drawbacks that are explained in the following and summarized in table 2.5.

Top-Projection

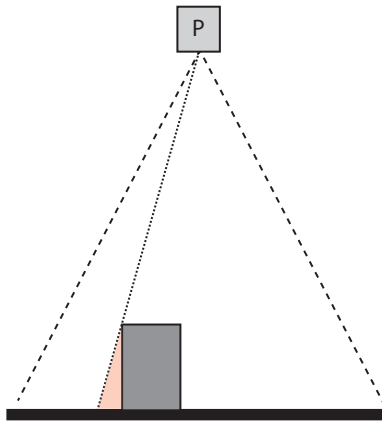


Figure 2.3: Shadowing caused by the user and tall objects: The red area is unreachable for the projector. If the user is sitting on the left side, a significant part of the visualization space is lost.

Top-projection uses a projection device mounted above the table that projects the image on the table surface (see figure 2.2a). In this context, the table can be basically any conventional off-the-shelf table. It only needs to have an opaque light-colored unstructured surface. Given the availability of affordable digital projectors, this is a very convenient and flexible approach. Besides installing the projector in a fixed setup on a wall or ceiling mount, it can even be mounted on a tripod and be installed or removed as needed. This flexibility makes the setup especially interesting for research, and hence top-projected tabletops are frequently used in projects [Cotting and Gross,

2006; Hardy, 2012; Kane et al., 2009] – the first implementation of a tabletop, the DigitalDesk by Wellner, also was a top-projected device [Wellner, 1993]. Implementing touch or pen input on the table is – for instance – possible using optical camera-based approaches, like stereo cameras [Agarwal et al., 2007; Kim et al., 2014] or depth cameras [Wilson, 2010]. Going one step further, interaction has even been extended to the back of top-projected tabletops

[Wigdor et al., 2006]. Considering hybrid settings, top-projection is not generally affected by the placement of objects on the table surface, since it simply projects on top of them.

However, top-projection also has some severe limitations and disadvantages. Since the projector needs to be mounted above the table, objects entering the space between the projector and table surface cause shadows. The most prominent “object” is the user himself, such as when the user bends over the table to casually read a projected document as if it were a printed document. Adding to this, tall objects present on the table cause shadowing as well – not only at the spot they occupy physically, but also on their surroundings, as depicted in figure 2.3. Other than that, projection quality is also affected by the properties of a surface like structure or color or ambient light (e.g., bright sunlight).

Back-Projection

Back-Projection is historically also implemented using a projector. For back-projection, the projector is mounted below the table, leading to the requirement that the table has a semi-transparent surface that is used for projection. Touch detection is often realized using optical techniques like FTIR (Frustrated Total Internal Reflection) [Han, 2005], Diffuse Illumination (DI) [Dang and André, 2011], or light curtains. Diffuse illumination has the advantage that it is additionally possible to detect fiducial markers on the bottom of objects that is invisible to the user. A major advantage of back-projection-based tabletops is that everything is contained in one convenient unit that can be easily transported. Since the image is projected from below the surface, the user can bend over the table or interact with his hand without interfering with the projection, which provides a more convenient user experience. While still relying mostly on standard hardware, it is now required to use a specially made table with an appropriate surface and a cabinet underneath that contains the hardware. This is necessary to provide adequate distance from the projector to the surface. Usually, a mirror is used to extend the path of the light within the enclosure, as shown in figure 2.2b.

Most commercially available devices sold in the past years, such as the well-known Microsoft Surface or Samsung SUR40 tabletop, are back-projected devices. The PixelSense technology used in the SUR40 made it possible to replace the projector with an LC-Display, eliminating the need for a bulky cab-

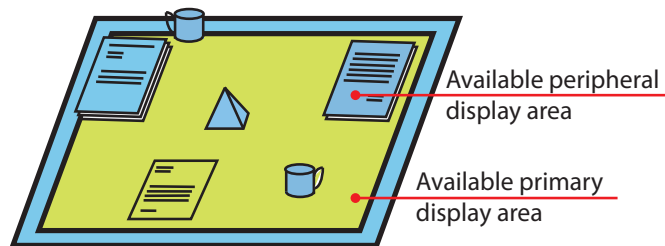


Figure 2.4: The two-class display concept uses the LCD to display digital content in high resolution (green) and provides the main interaction surface for the user. By means of projection, the surfaces of physical objects as well as the rims of the tabletop are augmented via projection to be used as secondary, peripheral display space (blue).

inet underneath the table (see figure 2.2c). To detect touch points and fiducial markers, a Diffuse Illumination-based approach is used with the light source and sensor pixels integrated into the LCD matrix. Recently, there is a tendency towards the use of capacitive touch sensing, which is completely immune to environmental light. However, this does not support marker recognition.

Despite all its advantages, back projection is especially prone to occlusion from objects placed on the table surface. Even a piece of blank paper, which a top-projected tabletop would project over, hides a substantial part of the display area which, as a result, becomes unusable. Also, there is no inherent awareness of the content that is underneath the occluding object.

2.4 PeriTop

Combining the properties of the two implementation styles of tabletops and the workspace model renders a two-class display hierarchy possible that combines the advantages of both top- and back-projection in an integrated environment. Thereby, we can provide a better user experience as well as foster a tighter integration of digital and physical content.

The basis for PeriTop is a back-projected tabletop that provides a high-resolution display surface in a form-factor that yields a good user-experience (desk-like size and height, comfortable to sit at, etc.). This serves as the *primary* interaction surface where users interact and conduct their work. As previously noted, the use of back-projection-based tabletops leads to severe occlusion by physical objects, leading to a significant reduction of screen space. To mitigate this, PeriTop combines the back-projected base device with an additional top-projection setup. The top-projection is able to leverage the



Figure 2.5: Envisioned Tabletop Setup with projector (green) / camera (yellow) add-on as a table lamp. The colored rectangles represent digital objects. Blue objects are shown on the tabletop's display and green ones via top projection.

surface of the occluder or otherwise passive areas, like the bezel of the tabletop or empty areas on physical documents, as extra display space. Since occluders are usually objects with surfaces that are not optimized for projection quality, they serve as a secondary display class, namely the *peripheral* displays (cf. Baudisch et al. [2001]). The name derives from the fact that they are mostly used to display “peripheral” information in the context of the PeriTop concept.

Peripheral information is information that is currently not in the focus of the user’s attention, but might be needed in the future and whose presence the user should be aware of (e.g., in the context of this thesis, occluded digital documents). As a result, a lower display quality is acceptable. This two-class display design avoids the inherent problems of working in only top-projected tabletop settings, such as large shadows cast by the user’s body or sensitivity to ambient light on the primarily digital workspace. Figure 2.4 depicts the two display classes.

Given the recent advances in tracking and projection technologies, together with the lower requirements for the peripheral displays, it is possible to use an inexpensive pico projector-depth camera pair as an add-on to digital tabletop systems that retains the integrated device aspect of back-projected tabletops as much as possible: The top-projection add-on can be placed on the table as a lamp-like device, as depicted in figure 2.5. While such devices have been prototyped in other research as well ([Kane et al., 2009; Rädle et al., 2014]),

	PeriTop	PeriTop Light
Primary Table Surface	LCD	Top-projection
Peripheral Displays	Top-projection	Top-projection
Object Tracking	Tabletop and RGBD-camera	RGBD-camera
Touch on Table Surface	Tabletop	RGBD-camera
Touch on Objects	RGBD-camera	RGBD-camera

Table 2.6: Technologies used in the two PeriTop setups.

they are meant to be used in the context of normal tables as a sole means of digital display and not as an add-on to tabletops that are already interactive by themselves.

2.4.1 Practical Implementations

The conceptual PeriTop setup described above is designed to deliver an optimal display quality by combining top-projection with an LCD-based tabletop. However, as outlined in the introduction chapter of this thesis, tabletops are currently not widespread in practical office environments. In order to address a wider range of application scenarios, some of the contributions in this thesis do not rely on the “full” *PeriTop* setup, but use a simplified hardware setup that omits the LCD-based tabletop and is completely top-projected. This reduced setup will be referred to as “*PeriTop light*” in the respective chapters. Generally, these setups are interchangeable from a conceptual perspective since they provide the same features, although with a different quality. For example, the use of top-projection instead of the LCD for the tabletop surface provides output on the table-surface as well, however with the issues of top-projection described above. Table 2.6 provides an overview of the technologies used.

2.5 Use Case Scenarios

Using our fundamental approach of leveraging two display classes, we can take a new design approach to the problems of occlusion and hybrid stacking. This fosters a tighter and more seamless integration of digital and physical documents. The following five exemplary use cases show how the peripheral display concept can be applied to keep UI elements visible (“*Stay-On-Top*”), expand the usable surface for storage of documents and interaction to the rims of the tabletop (“*Off-Screen-Storage*”), resolve occlusion caused by physical

	Physical			Digital	
	Tangible	Document	Other	UI	Document
Stay-On-Top	○	●	●	●	○
Off-Screen Storage	○	○	○	○	●
Occlusion Resolution	○	●	●	○	●
Hybrid Stacking	○	●	○	○	●
Tangibles	●	○	○	●	○

Table 2.7: Overview of the use case scenarios with regard to the types of objects covered primarily.

objects (“*Occlusion Resolution*”), provide the possibility to arbitrarily stack digital and physical documents (“*Hybrid Stacking*”), and integrate interactive tangibles (“*Tangibles*”). The use cases were chosen to cover all object types identified in section 2.3.2 and cover a wide range of practical application scenarios. Table 2.7 shows the relation between the use cases and object types.

While the first two applications are rather straight forward use-cases, the latter three pose a set of challenges and are hence explored in-depth in the next two parts of the thesis.

2.5.1 Stay-On-Top

In 2D desktop environments, objects like widgets, tool palettes, or other application controls always stay on top of all other content or application windows. This facilitates the user’s access to these objects regardless of the current task at hand. Despite the important role of the controller widgets, they are normally not in the user’s primary focus (for instance, the user only uses them to switch between tools). Due to the presence of physical objects in hybrid tabletop settings, such widgets can get easily occluded or become difficult to access by the user.

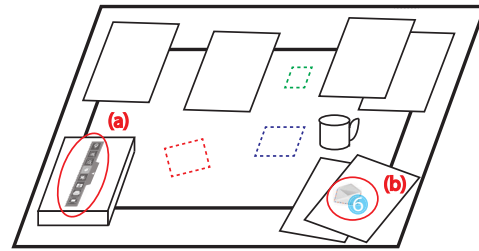


Figure 2.6: a) The tool palette is top-projected on top of the book. b) An e-mail notification icon is shown on the paper stack.

By leveraging the PeriTop concept, control widgets like a tool palette can always be visible to the user by projecting them on physical objects placed on the tabletop surface. Furthermore, the tool palette can be moved off the screen to the tabletop’s rim in order to have more screen space available for

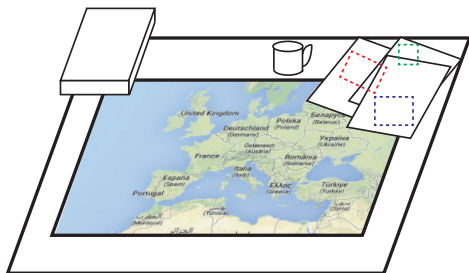


Figure 2.7: Peripheral ad hoc off-screen storage of objects (dashed rectangles represent digital objects).

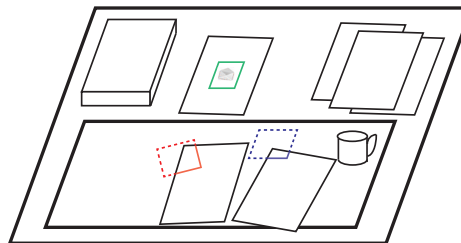


Figure 2.8: Only the primary working area of the table is equipped with an LCD.

interacting with the main content (see figure 2.6).

Another concrete example of hybrid stay-on-top is system or application notifications. In 2D desktop environments, notifications appear on top of the other content/windows, so the user can immediately perceive them. Unrestricted placement and tall physical objects on the tabletop surface can block the user’s perception of important notifications. Although relocating notifications to non-occluded areas of the tabletop surface can mitigate the problem, it is not practical for massively cluttered settings or for notifications that should pop up at predefined positions on the screen (for instance, email, calendar, or application-related notifications).

Leveraging the peripheral display concept in PeriTop, notifications can be projected onto physical objects at or as close as possible to their predefined and original locations on the screen. This ensures users are immediately aware of them. Furthermore, they can then drag them to a convenient free tabletop screen area to see more details.

2.5.2 Off-Screen Storage

Knowledge workers frequently switch between different tasks. In order to change the context of a task in traditional office settings where users work with a computer desktop and a number of printed documents, notepads, or books, they normally first clear away or simply put aside the current task’s materials. After making sufficient free space in the working area of the desk, users can start the new task.

A study by Steimle et al. [Steimle et al., 2010b] showed that users store digital and physical documents in the form of hybrid piles on interactive tabletops where digital documents are stored under a pile of physical documents. They

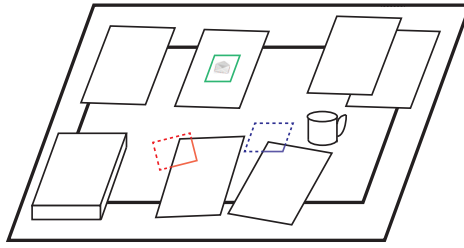


Figure 2.9: Physical objects used to resolve occlusion: Digital objects get visualized at their true position (in occluded areas, only the outline and, if the object is hidden more than half, an icon are projected).

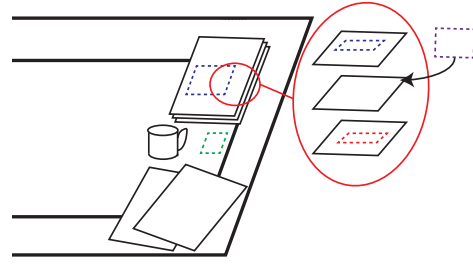


Figure 2.10: True hybrid piling: Digital Objects (dashed rectangles) can be put at any position in-between physical objects.

identified that both hybrid storing and retrieving of documents are cumbersome activities.

Peripheral displays on physical objects or tabletop rims can act as off-screen storage areas, thus allowing users to put aside digital content together with the physical objects and retrieve it later. The advantage over minimizing them on the primary display is that the objects stay visible to the user and can be retrieved quickly (see figure 2.7).

Further, instead of using the rims of an all-display tabletop, it is also possible to only augment the main working area of a table with an active display and rely on the peripheral displays for the storage area (see figure 2.8).

2.5.3 Occlusion Resolution

Traditional approaches resolve occlusion using display space next to the occluder to provide awareness of and access to the occluded objects [Javed et al., 2011], which requires even more display space. The PeriTop concept allows to leverage the surface of an occluder for this purpose (see figure 2.9); thereby screen space is conserved.

2.5.4 Hybrid piling

While storing away or grouping documents, piling of the collection of documents to be stored is a common practice. By leveraging the PeriTop concepts, users can be enabled to arbitrarily interweave digital and physical objects at any position in a pile (see figure 2.10), enabling true hybrid piling while not limiting digital objects to be either below or on top of the physical part.

2.5.5 Tangibles

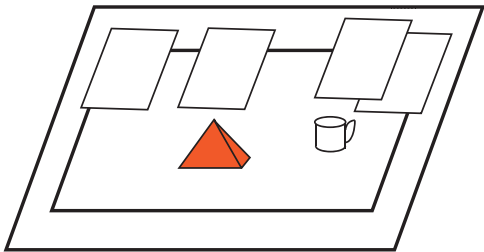


Figure 2.11: Texturization of a tangible pyramid.

Leveraging the concept of additional top-projection renders visual output on arbitrary objects feasible. Besides projecting smaller pieces of information, it is also possible to completely texturize the surface of an object. This can be leveraged to implement visual output on tangible objects (see figure 2.11); the touch detection can

be used to provide true interactivity by means of multitouch support.

2.6 Implementation Issues

For our research, we did not use the integrated lamp-like device proposed in figure 2.5 (section 2.4, page 35), but a non-integrated setup instead to facilitate hardware adjustments (e.g., re-aligning the cameras). The setup consists of a Samsung SUR40 tabletop running Microsoft Windows as the primary device. The SUR40 not only provides a Full-HD (1920×1080 pixels) display, but also touch recognition and fiducial marker tracking. Additionally, the raw image provided by the PixelSense technology is available to perform custom object tracking. The raw image is used to recognize occluding objects even without markers by means of computer vision, as described later

To top-project and enable multitouch input on non-display areas, a standard Full HD projector and a Microsoft Kinect depth camera (later replaced by a Kinect v2) were mounted above the SUR40 (see figure 2.12). In order to detect physical objects and subsequently touch events on their surface, a modified version of the dSensing NI approach [Klompaker et al., 2012] is employed. The reasons for the modifications and the modifications themselves are described in the next subsection.

The resulting system must cope with two touch sources – the tabletop itself that delivers the events via the Windows touch API and the depth camera framework that provides them via the TUIO protocol. In order to avoid false or double touches, the events are first transformed into a unified coordinate system and then preprocessed to remove touch events that are reported by the “wrong” channel for a given area. To do so, the PixelSense raw image is



Figure 2.12: Physical objects causing occlusion of digital documents. The occlusion is resolved in situ by using an outline plus icon visualization projected onto the occluding objects.

used: In areas that are covered by objects and hence occluded, touch events by the touch API are rejected, while in other areas events received via TUIO are rejected. To further account for double detections at the border areas of objects where one would be detected on the object and one next to it, touch points that are too close together are also rejected.

2.6.1 IR Interference

Both the Microsoft Kinect and the SUR40 tabletop operate using infrared light. The SUR40 illuminates the whole display surface to detect objects by sensing the light reflected back. The first Kinect uses a structured light pattern (see figure 2.13) that is projected via an infrared laser to measure depth by analyzing the deformation of the pattern.



Figure 2.13: The structured light pattern used by the Kinect.

Hence, there is a strong interaction between these two devices: The SUR40 detects the bright dots of the structured light pattern from the Kinect as touch events, which leads to tens of random touch points that even exceed the maximum

number of touch points supported by the SUR40. Further, the massive number of touch-points causes performance issues, rendering the entire touch detection functionless. In the other direction, the Kinect is “blinded” by the SUR40 and hence incapable of measuring the depth of the display surface.

To solve the false touch point issues, it was sufficient to mount the Kinect at a more flat angle. As a result, the light from the Kinect is mostly reflected by the plastic surface of the display and no longer causes touch events. The minimal number of still occurring events can easily be filtered because they are smaller than a typical finger.

In the other direction, the Kinect remains blinded by the SUR40. Since dSensingNI relies on background subtraction to remove the static background, it did not function under these conditions. The reason is that the Kinect reports an infinite distance for the surface of the tabletop with some smaller clusters of random values. As a result, it failed at detecting objects and subsequently touch input on their surfaces. To compensate for this, we reimplemented the dSensing NI approach based on the description of the paper with added handling for invalid regions. If an area with a size above a threshold and sensible depth values appears within the previously invalid area, it is also considered as an object. Because there is no reference depth value for the tabletop surface, the height of the object is not directly available and must be calculated later by using a static reference plane that models the surface of the tabletop. The plane is determined by placing a piece of cardboard on the tabletop during calibration to allow the Kinect to sense the corresponding depth-values. However, this reference plane cannot be used directly for background-substraction since it does not model the objects on the tabletop or the depth offset caused by the sensing approach used by the Kinect, which might vary depending on the position within the field of view of the Kinect. In order to allow stacking of objects and touch on movable objects, a background map without the table surface was maintained as described before and updated continuously. The reference plane is only used to determine the height of these objects.

2.6.2 Object/Document Recognition

In many scenarios (e.g., to provide hybrid stacking), it is not only relevant to know whether an object is placed on the surface, but also to identify the object.

A common approach is to use fiducial markers that represent an unique

identifier. They are then printed or glued onto the objects of interest. Often, the markers are recognized by a camera mounted above the table. A major drawback in this case is that the markers are visible to the user and can be distracting. Research has investigated the use of IR ink to print invisible markers [Willis et al., 2013]; however, this is a complex process and not suitable to quickly augment – for instance – existing documents or books. Given the use of a SUR40 tabletop, conventional markers could also have been mounted on the bottom of the objects, invisible for the user. However, this would have limited the recognition to the bottommost object in case of multiple being stacked. Further, the use of markers requires to augment all objects with markers.

Instead of following that line further, we chose another common way for object recognition. Since a Kinect is mounted above the table for touch recognition, the RGB camera video feed is used to detect objects using SIFT-features [Lowe, 2004]. Depending on the size of the collection of known documents, this can be a time consuming process, since the positions of the documents within the camera feed are usually unknown. To improve performance, the raw PixelSense feed from the SUR40 is used to identify the positions of documents: The PixelSense technology provides an infrared image of the desk surface taken from below in which covered areas are brighter due to the light reflected from the object. Using blob and corner detection, the contours of these objects can be extracted. Since the camera is calibrated relative to the tabletop, these contours can be used to clip the document regions in the kinect video feed, leading to precisely cut out objects. The SIFT-based recognition is then executed on the smaller regions. This approach also solves the problem that digital documents on the tabletop display are recognized as well since the camera cannot easily distinguish between a physical document and a digital one on the tabletop.

Since we are also interested into stacking objects, this approach has further advantages: As the camera films the scene from the top, it can also recognize documents added later on top or detect the removal of documents when it recognizes again a document previously seen at a given location. Thereby, a digital model of a stack can be built. To avoid interference through the projection onto the documents, the projection can be briefly disabled if a document is added or removed.

2.6.3 Calibration

In order to realize the PeriTop concept of peripheral displays, the setup requires the calibration of the various data sources and display technologies relative to each other. To do so, the tabletop is used as a coordinate reference so that the screen is aligned in the x-y plane at $z = 0$, with the lower left point of the screen having the coordinate $(0, 0, 0)^T$ with the z-axis pointing upwards. The Kinect and the projector are then calibrated within this reference coordinate system. To have an intuitive unit of measurement that is device independent, the coordinates are internally expressed in mm. By defining the table surface to be at $z = 0$, the z-coordinate of an object in the digital model can be easily used to decide whether it is shown on the tabletop display or via projection: positive values mean projection; negative that it is to be displayed on the tabletop.

To calibrate RGB cameras together with projectors, many computer vision libraries provide checkerboard-based automatic calibration tools that yield a suitable transformation matrix. As a result of the abovementioned requirements on the coordinate system, we chose to implement a reference point-based approach based on work by Zhang [1993/2010]. It allows calibrating the camera and projector independently within the coordinate system described above. To do so, the approach requires more than five point correspondences to be known (i.e., the coordinate within our reference system and the corresponding coordinate in the camera or projector coordinate system). For our implementation, we chose to use at least eight for increased accuracy.

For the camera, the depth data directly delivers a 3D coordinate so that the calibration is done by first touching a set of four points on the tabletop surface. Due to the problem of IR-interference discussed previously, a sheet of paper needs to be placed on the tabletop to get a coordinate-measurement. Since the point references need to be non-coplanar, an additional set of four points that are above the table surface need to be recorded. To do so, we used a box of known height (in our case 15cm) that was placed at the reference points and touched. The reference points are indicated by dots on the tabletop display during the process. Since the RGB camera is mounted in the same assembly as the depth camera, the known intrinsic transformation for the Kinect can be used to calculate the transformation for the RGB camera based on the calibration results from the depth camera. While this is not as precise as directly calibrating it, the result is sufficient for recognizing documents and saves an additional calibration step.

The reference points for the projector are recorded similarly (four on the display surface and four above). To do so, a reference point is displayed on the tabletop and the user has to move a projected point onto the reference point. The projector coordinate system is given by the fact that the projector is internally modeled as a (DirectX) camera that uses a coordinate system from -1 to 1 for the x - and y -coordinates and 0 to 1 for the z -coordinate (0 being closer to the camera). For example, the tabletop displays a reference point at $(50, 50, 0)^T$ (5 cm away from the lower left corner on the tabletop surface). To match the point, the user has to move the calibration point in the projected image (that is 1920×1080 pixels in size) to the 2D pixel coordinates $(100, 300)^T$. This results in the projector coordinates $(100/1920 \cdot 2 - 1, 300/1080 \cdot 2 - 1, 1)^T = (-0.896, -0.444, 1)^T$.

In order to compute the transformation matrices, the problem is transformed into an Eigenvector problem [Zhang, 1993/2010] that is solved using a Rayleigh quotient iteration in our implementation. The resulting Eigenvector belonging to the smallest Eigenwert is then transformed back, yielding the transformation matrix.

At the cost of manually defining the reference points, it provides a consistent calibration approach that can be used for depth camera data and touch points as well. Further, it is possible to refine the points to adjust the mapping manually. To increase resilience against noisy measurements, the approach further scales directly to more than the eight reference pairs that are currently used. The RMS error of this approach is usually less than 1 mm with careful capturing for the projector and less than 5 mm for the touch input.

2.7 Summary and Conclusion

In this chapter, we presented PeriTop, a novel two-class tabletop display concept and framework for hybrid tabletop environments that builds the foundation for the subsequent contributions made in this thesis. The PeriTop concept combines the advantages of top- and back-projected tabletop systems and enables the use of the surfaces of physical objects as additional peripheral display space. By means of depth-camera-based touch sensing, the peripheral displays cannot only be used as a passive display, but also as an interactive surface. However, due to technical reasons, the touch input is currently limited to surfaces mostly parallel to the desk surface, i.e., touches on the sides of an occluder or steeply slanted surfaces are not recognized. Further, we presented

a first set of use cases to show the benefits of the concept of peripheral displays and three of which are further explored in the following chapters. The chapter concluded with a description of important issues of the actual implementation of the PeriTop concept.

Part II

Stationary Hybrid Knowledge Work

ProjecTop – Resolving Occlusion in Situ

Occlusion is one of the major issues in the context of hybrid workspaces. This chapter explores the use of in-situ occlusion resolution in hybrid tabletop environments by introducing the ProjecTop concept. Unlike the current state of the art, ProjecTop resolves occlusion without the need to use additional tabletop display space by leveraging the PeriTop concept described in chapter 2. Further, since occlusion is resolved in-situ, the spatial layout of the occluded objects is preserved.

After an introductory section, the chapter starts with a review of relevant related work from the field of occlusion resolution (section 3.2). Section 3.3 then presents an initial study to inform the design of ProjecTop and based on its results, a set of visualization and corresponding interaction techniques for in-situ occlusion resolution and easy access to occluded objects is proposed in section 3.4. The ProjecTop techniques are then evaluated in a user study presented in section 3.5 and discussed in section 3.6. Based on the discussion of the results, we present a set of design implications for in-situ occlusion resolution and the use of peripheral displays in section 3.7. The chapter concludes with a summary of the contributions in section 3.8.

3.1 Introduction

As outlined in the motivation of this thesis, occlusion is a prevalent problem in hybrid tabletop environments. In this chapter, we investigate the use of peripheral displays to resolve occlusion in situ on the surface of occluding objects instead of next to them on the tabletop display. The latter is the current state of research for approaches providing awareness for occluded objects. Nevertheless, this approach has two major drawbacks:

First, the visualization next to the occluding object requires display space

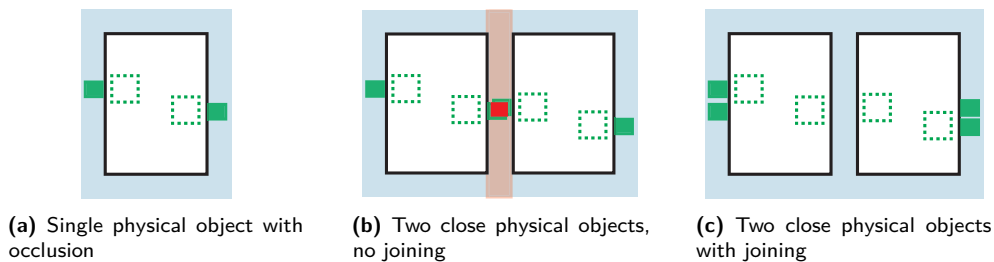


Figure 3.1: For single physical objects, visualizing occluded objects next to the object using proxies (green rectangles in the blue area) is no problem. With two close-by objects, the visualization areas (light blue) as well as the proxies overlap (highlighted in light red), which causes usability issues. Joining the two objects into a single cluster resolves the collision, however, the proxies are farther away from the actual position and the actual spatial layout becomes unclear.

on the tabletop. In scenarios with only a few occluding objects present, this does not pose a problem. However, with an increasing number of objects, there might not be enough space next to the occluding object to show the visualization. As a result, the visualization would have to be moved even farther away.

This leads to the second issue, namely that the spatial layout cannot be understood when the visualization is shown too far away from the actual position of the represented object. See figure 3.1 for an example: While for a single physical object, the visualization space around the object (blue shaded area) is free to display a proxy representation of occluded objects (objects in dashed green, proxies in solid green), two close-by physical objects lead to problems: either the visualization spaces as well as (in adverse cases) the proxies overlap – which leads to display as well as interaction problems – or the two occluding objects are “merged” into an “occluder group” and the visualization is moved to the border of the group, thereby losing the spatial perception. Of course, the example could be extended to even more close-by objects, which would increase the problem.

To overcome these issues, we contribute a set of integrated interaction and visualization techniques that leverage the PeriTop concept presented in the previous chapter. ProjectTop uses the peripheral display concept for in-situ occlusion resolution in hybrid environments. The design of our techniques is grounded on an initial user study. All presented techniques are coherently implemented in a fully functional occlusion support system, named ProjectTop as well, that allows the users to fluidly interact in a realistic hybrid tabletop setting (see figure 3.2).

Using the implemented system, the ProjectTop concept is evaluated in a user



Figure 3.2: ProjectTop system visualizing the occluded digital objects.

study in a real hybrid setting. The goal was to assess the performance of the developed techniques for occlusion management in hybrid settings. Based on the results, we present design implications for the future development and improvement of hybrid tabletop systems extended with top-projection. We believe the results are not only relevant in the context of occlusion but also in general for the use of top-projection to project additional self-contained content on physical objects.

3.2 Related Work

The work presented in this chapter relates mainly to two research fields – namely augmented desktops and occlusion – which are discussed in the following sections.

3.2.1 Augmented Desktops

Since ProjectTop relies on the PeriTop concept to project onto physical objects, the related work on augmented desktops is also highly relevant for this chapter. A detailed discussion of specific works is, for redundancy reasons, omitted here. For details, refer to section 2.2.1.

As previously discussed in the aforementioned section, existing works often lack support for multiple concurrently used physical documents and self-contained digital documents, especially in combination. As a result, the problem of occlusion is usually not taken into account in these approaches since there is a strong focus on only one type (digital or physical) of documents rather than on their combination. Further, multitouch has been established as a common input modality. Thus, multitouch without additional tools should

be supported. By using PeriTop as a basis for ProjecTop, these drawbacks are circumvented.

3.2.2 Digital Occlusion

Occlusion of digital content is not a problem that emerged with the advent of tabletops. It was already explored in purely digital scenarios: With the advent of multiple overlapping windows on computers, occlusion of digital objects by other digital objects became a relevant issue. As a result, research has emerged to address this problem.

A common starting point is the compositing process for the digital interface: Usually, the different windows are rendered opaque and hence wholly hide any other window that is underneath. This is no problem if the user is focused on the content displayed in the topmost window(s) – however, with the advent of complex applications that use stay-on-top tool palettes, screen space becomes limited and the actual content is occluded by the palette windows. As a result, the use of transparency for such overlay windows has been explored [Harrison and Vicente, 1996; Harrison et al., 1995], and it was shown that transparency is to a certain degree beneficial for awareness of the underlying content without affecting user performance when interacting with a menu. However, for transparency values greater than about 50%, the user performance starts to degrade due to visual interference between the menu and the underlying content. This is also dependent on the structure and texture of the menu (text or icons) as well as the background (text, solid color, structured color, etc.). To address these issues, Baudisch and Gutwin [2004] developed a multiblending technique that combines different compositing methods such as embossing, alpha blending, or cropping depending on the visual feature. The resulting glass-style palette windows could significantly reduce the error rates compared to alpha blending-based approaches while at the same time providing a better visual reproduction of the occluded content.

In another direction, for complete windows, context-aware transparency was explored [Ishak and Feiner, 2004]. This approach leverages the fact that windows often have large unused areas (e.g., the white space in a text editor), that could be made transparent. Instead of making the whole window transparent, which would lead to the same interference problems as above, the regions that contain, for instance, text, are not made transparent. As a result, the windows can be transparent while at the same time their content remains

legible. Going one step beyond making windows transparent, Waldner et al. [2011] developed a complete window management solution that is importance-driven and optimizes the visibility and interactivity of occluded window content. In addition, the layout of the windows is semi-automatically adjusted to further optimize the accessibility and visibility of important content. A first user study showed that this approach is more effective and satisfactory than transparent windows.

While inspiring, these approaches cannot be directly transferred to hybrid scenarios. Although it is easy to achieve transparency of a set of digital documents through advanced composition techniques like alpha-blending [Porter and Duff, 1984], physical documents are not capable of selectively changing their transparency. However, using projection, a similar effect can be achieved in the physical world as well as in hybrid settings. Since this requires the combination of different display techniques (ink on physical paper and projection), it introduces a more notable visual break compared to a purely digital setting. Nevertheless, transparency is an important means for providing awareness or access to content without having to relocate it. Hence, a similar concept will be used in the context of ProjecTop.

Along another line, Geymayer et al. [2014] presented an approach that does not use alpha-blending or other specialized composition techniques, but visual indicators. The approach is built around search processes, where the occurrence of a keyword is highlighted in the visible windows as well as currently occluded windows using an overlay that shows a box around the area where the information is (independent of whether it is currently visible or not) as well as lines connecting the boxes. Arrows indicate even content that is outside the screen (e.g., due to window placement). However, this approach is not generally applicable since it requires a keyword to search for, which is not usually possible for other visual content, such as images. Tumble! Splat! by Ramos et al. [2006] solves the problem of awareness and access for digital/digital occlusion in drawing applications by temporarily spreading out the objects that are otherwise on top of each other in the x-y plane, thereby providing easy access to them.

3.2.3 Physical Occlusion on Tabletops

Occlusion by physical objects has been widely studied as it is a common problem even in physical-only settings. Iwai et al. developed the limpid

desk [Iwai and Sato, 2006], which allows for ad hoc transparentization of stacks of physical documents to browse them. Underlying documents are projected on the topmost document if the user wants to look at them. However, they investigated a physical-only setting, in which the presence of occluded documents is perceivable without the need for any visualization. As a result, the system needs to project only when the user requests to see an occluded document, but does not need to continuously provide awareness of the occluded documents. Our work extends this approach to digital items on a hybrid interactive tabletop. Steimle et al. [2010b] as well as Khalilbeigi et al. [2010] conducted studies to gain an understanding of how physical-only interaction transfers to mixed digital-physical interaction scenarios on interactive surfaces. They found that concepts common in the physical world, like stacking, are also applied to hybrid settings by the participants. Therefore, the problem of physical/digital occlusion can be assumed to be an important factor in the usability of such systems.

In order to completely avoid occlusion, systems have been developed that adapt the digital display space to align around physical obstacles. Most notable are the interactive environment-aware display bubbles by Cotting and Gross [2006]. They used a potential field-based approach to calculate bubble-shaped areas flowing around physical objects present on the desk, thereby avoiding occlusion of the digital content. However, this limits applicability for smaller areas between physical objects since the bubbles do not expand into small gaps. A simpler approach was proposed by Freeman and Brewster [2013], who developed a concept for finding free space on tabletops in order to avoid occlusion by moving objects to large-enough free areas. In another direction, user involvement has been used to avoid occlusion by aligning menus along a user-drawn path [Leithinger and Haller, 2007]. Here, the user avoids obstacles when drawing the path and the menu items can be displayed at a convenient location without being occluded or interfering with physical objects.

Avoiding occlusion is working fine in settings with much free space available, and the objects need to be moved only a bit. However, in highly cluttered environments, where no nearby free space is available, it leads to significant reorganization of the user’s workspace, which is undesirable. Other research [Tabard et al., 2013] also suggests that users do not need any reactive occlusion support, as (unwanted) occlusion seldom occurs. This is mainly because users arrange their workspace in advance to avoid it or the system can do it by relocation. However, the system used in the study did not provide any occlusion

support; therefore, users might have avoided occlusion by themselves due to the lack of support rather than because they avoid it in general. Also, the setting of a lab bench is rather special and probably not representative for, e.g., usual office work. Besides ways to avoid occlusion, there are concepts to deal with it: Systems like SnapRail [Furumi et al., 2012], occlusion management techniques by Javed et al. [2011], or ObjecTop [Khalilbeigi et al., 2013] provide means for the user to interact in hybrid settings without losing awareness of occluded objects but without limiting the user, for example, by relocating occluded items to avoid occlusion. This is done by using interactive proxy objects, e.g., tiny icons representing occluded objects on unoccluded space, allowing for access and perception. However, while helpful for the user, these techniques all consume additional display space on the tabletop. This is not an issue as long as only a small amount of the display is covered, but it might become a problem if there is not much display space left to display the visualization.

3.2.4 Occlusion by the User’s Body

Besides occlusion due to digital or physical objects, the user can also cause occlusion through his body. Hands and arms especially cause occlusion when interacting with a touch screen. However, since the user likely moves his arms around during interaction, the occlusion is not as permanent as occlusion caused by other objects. Contrary to occlusion by physical or digital objects, the user’s hands are always where the user wants to interact, which might lead to problems when covering, for instance, menus. As a result, research has modeled the areas occluded by hands on small displays like tablets [Vogel et al., 2009] or larger interactive tabletops [Vogel and Casiez, 2012]. To avoid issues during interaction, menu concepts were developed that spare the areas covered by the hand and lay out the menu for convenient access [Brandl et al., 2009] or even complete interfaces that adapt to the occluded areas by means of callout bubbles [Vogel and Balakrishnan, 2010]. Along another line, the occlusion caused by the user’s arm has been used to trigger menus or windows that are bound to the user’s arm [Koura et al., 2012] or even used the arm itself to display menus and thereby circumvent occlusion [Adachi et al., 2013]. Also, the use of the back of the display has been explored to provide occlusion-free interaction for small-scale displays [Wigdor et al., 2007] as well as tabletops [Wigdor et al., 2006]. However, this stream of research is of lesser importance to this work and is only mentioned for completeness since we focus primarily on occlusion caused by physical objects.

3.2.5 Requirements and Summary

In the following, we present a set of requirements for occlusion support in the context of hybrid environments derived from reviewing the related work. The requirements are then used to compare the previously discussed related work (see table 3.1).

- R1 **Support for digital documents** For application in hybrid scenarios, self-contained digital documents (e.g., images, text documents, e-mails, etc.) need to be supported.
- R2 **Support for physical objects** For application in hybrid scenarios, physical objects need to be recognized as well to react to occlusion caused by them.
- R3 **Awareness of presence** In case of occlusion, the user should be made aware of the – otherwise hidden – documents.
- R4 **Awareness of layout** In order to facilitate access to occluded documents, the spatial layout of occluded documents should be conveyed.
- R5 **Awareness of type** To facilitate finding relevant documents, the type of an occluded document (e.g., e-mail, text, image) should be indicated to the user.
- R6 **Easy access** Interaction techniques to grant easy access to occluded documents should be provided.
- R7 **Allow occlusion** Since occlusion is an important means of organization, it should be possible for the user to leverage when desired. Therefore, systems should not make it impossible to occlude a document, e.g., by relocating it automatically to a free spot.

3.3 Initial Study

In the context of ProjectTop, we are going to use the surface of occluding objects to resolve occlusion in situ. In order to gain an understanding of suitable forms of visualization for the in-situ visualization of occluded objects, we conducted an initial user study comparing three different visualization styles that is described in the following.

	Support for Digital Documents	Support for Physical Objects	Awareness of Presence	Awareness of Layout	Awareness of Type/Content	Easy Access	Allow Occlusion
<i>Digital Occlusion</i>							
Harrison et al. [1995]	●	○	●	●	●	○	●
Baudisch and Gutwin [2004]	●	○	●	●	●	○	●
Ishak and Feiner [2004]	●	○	●	●	●	●	●
Waldner et al. [2011]	●	○	●	●	●	●	●
Geymayer et al. [2014]	●	○	●	●	●	●	●
Ramos et al. [2006]	●	○	○	○	○	●	●
<i>Physical Occlusion</i>							
Iwai and Sato [2006]	○	●	●	●	●	○	●
Cotting and Gross [2006]	●	●	○	○	○	○	○
Leithinger and Haller [2007]	●	○	○	○	○	○	○
Tabard et al. [2012]	●	●	○	○	○	○	●
Furumi et al. [2012]	●	●	●	○	●	●	●
Javed et al. [2011]	●	●	●	●	●	●	●
Khalilbeigi et al. [2013]	●	●	●	●	●	●	●
<i>Occlusion by the User's Body</i>							
Vogel and Casiez [2012]	●	●	○	○	○	●	●
Brandl et al. [2009]	●	●	○	○	○	●	○
Adachi et al. [2013]	●	●	○	○	○	●	●
Wigdor et al. [2006]	●	●	○	○	○	●	●

Table 3.1: Properties of selected systems dealing with occlusion. ● indicates that a requirement is fulfilled, ● indicates partial fulfillment, and ○ that a requirement is not fulfilled.

	Low	Mid	High
Position	●	●	●
Orientation	○	●	●
Object Type	○	●	○
Identification	○	○	●

Table 3.3: Information conveyed by different visualization levels (Low, Mid, and High).

3.3.1 Study Design

<i>Method:</i>	Quantitative
<i>Participants:</i>	8
<i>Duration:</i>	ca. 20 min
<i>Data Gathering:</i>	Questionnaire, interview
<i>Variables</i>	
<i>Independent:</i>	Type of visualization
<i>Dependent:</i>	Questionnaire

Table 3.2: Study design overview.

The ProjectTop concept approaches the problem of occlusion by using the surfaces of physical objects as additional displays. In order to appropriately project digital content on complex surfaces with various textures, we conducted a user study with 8 participants (2 female, aged 26.5 years on average). The goal of this study was

to find a representation of occluded objects that conveys as much information as possible while being acceptable to the user. In particular, we wanted to see which properties of the underlying objects are important to the user. We therefore employed three visualization levels showing different characteristics of the represented object: Low (location only), Mid (location, orientation, and type), and High (location, orientation, and content). For a summary, see table 3.3.

3.3.2 Apparatus and Layout

The study was conducted using the PeriTop setup described in section 2.4. To implement the three visualization levels, we chose a small red dot (the most minimalistic visualization possible) for the “Low” level, and, based on related work [Javed et al., 2011; Khalilbeigi et al., 2013], an icon for the “Mid” level and a miniature version (thumbnail) for the “High” level.

These visualizations were projected on a set of physical documents consisting of three different types: a document containing only text, a document containing text with a large half-page image, and a personal notepad for handwritten

Variable	F-Value	Significance
Perceived Clutter	$F_{2,14} = 19.21$	$p < 0.001^*$
Digital Interference	$F_{1,2,8,2} = 14.04$	$p = 0.001^*$
Physical Interference	$F_{2,14} = 0.07$	$p = 0.84$
Acceptable (working area)	$F_{2,14} = 11.47$	$p = 0.001^*$
Acceptable (storage area)	$F_{2,14} = 2.44$	$p = 0.15$
Usefulness	$F_{2,14} = 22.53$	$p < 0.001^*$

Table 3.4: F-statistics for the influence of the visualization type. * denotes significant influence.

notes. To see whether the location of the object mattered, we distributed them across the table so that there were some within the working area of the user and some within the storage area. This led to a total of three trials per participant (one for each visualization).

3.3.3 Task

A set of documents with top-projected content was presented to the participants. In order to force engagement with these documents, the participants had to answer questions related to the content of the documents. To find out whether top-projection obstructs with user interaction with documents, the questions had to be answered in writing on a notepad that also had top-projected content on it. The set of documents was changed for each trial. The study lasted about 20 minutes per participant. After each trial, the participants filled in a questionnaire to rate the visualization and placement presented in the trial regarding the perceived clutter, usefulness, and interference between physical and digital content (each on a 5-point Likert scale), as well as acceptability of the projection (0 “never” to 10 “always”) in the working and storage areas. The interference between physical and digital content was split into two items: digital interference – “the digital content causes interference for the physical content” (e.g., the user wants to read the physical document under the projection); and physical interference – “the physical content causes interference for the digital content” (e.g., the user wants to recognize the digital content projected onto a physical document).

3.3.4 Results

We performed a repeated measures ANOVA on the data. Greenhouse-Geisser correction was used in case of violation of sphericity.

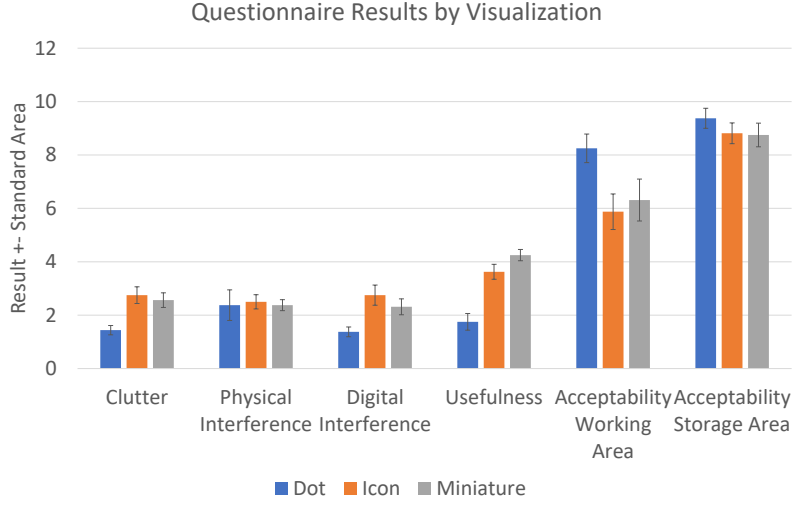


Figure 3.3: Questionnaire results by visualization.

As we expected, the type of visualization had a significant influence on the perception of the interface. When looking at the different items of the questionnaire, there was a significant influence on the perceived clutter ($F_{2,14} = 19.21$, $p < 0.001$), the digital interference ($F_{1,2,8,2} = 14.04$, $p = 0.001$), the helpfulness ($F_{2,14} = 22.53$, $p < 0.001$), and the acceptability of projection in the working area ($F_{2,14} = 11.47$, $p = 0.001$). There was no significant difference for the acceptability in the storage area ($F_{2,14} = 2.44$, $p = 0.15$) or the physical interference ($F_{2,14} = 0.07$, $p = 0.84$). Table 3.4 summarizes the F-statistics.

As expected, the dot visualization introduced minimal clutter ($\mu = 1.4$, $SE = 0.175$) and digital interference ($\mu = 1.3$, $SE = 0.18$), however, it was not very helpful ($\mu = 1.75$, $SE = 0.31$).

The icon and miniature visualizations were rated very similarly and introduce more clutter (miniature: $\mu = 2.75$, $SE = 0.31$; icon: $\mu = 2.56$, $SE = 0.27$), but were perceived as much more useful compared to the red dot (icon: $\mu = 3.63$, $SE = 0.28$; miniature: $\mu = 4.25$, $SE = 0.21$).

Post hoc analysis revealed that the difference between icon and miniature is not significant (all $p > 0.05$). However, the differences between dot and miniature as well as icon are significant (all $p < 0.016$).

Depending on the location, the acceptability varied. In the working area, the red dot received an average score of 8.25 ($SE = 0.535$), the icon 6.313 ($SE = 0.785$), and the miniature 5.875 ($SE = 0.666$). While the acceptability is clearly in favor of the red dot, the icon and miniature still receive scores on the more acceptable side (> 5). Post hoc analysis revealed that the differences

between icon and miniature are not significant ($p = 1.0$); only the dot differs significantly from the miniature ($p = 0.009$) and icon ($p = 0.016$).

Although not significant, the red dot was also ranked as most acceptable in the storage area ($\mu = 9.375$, $SE = 0.375$) followed by the miniature ($\mu = 8.813$, $SE = 0.389$), and the icon ($\mu = 8.75$, $SE = 0.443$).

All three visualizations received values above 5.8 for acceptability in the working area and above 8.7 for the storage area, leading to the conclusion that they are generally all acceptable for most users.

User Feedback

While the dot has certainly been rated as very acceptable in both the working and storage areas, it has also been found not to be very useful by the users (e.g., P4: “*Points were essentially useless*”, P6: “*Points are not so good since they are just red*”). The icons and miniature, in contrast, were seen as generally positive since they reveal more information about the hidden documents (e.g., P8: “*Miniatures are very helpful*”, P1: “*Icons in the storage area are extremely cool*”). P6 suggested adding an outline to the icon in order to better convey the size of an object.

3.4 Projec^Top

Based on the results of the initial study, we developed a set of visualization techniques for hybrid occlusion resolution using top-projection. Further, complementary interaction techniques were required to accommodate the fact that interaction occurs across the boundaries of an object (e.g., moving a digital object away from underneath an occluding object). We will describe both in the following.

3.4.1 Representing Digital Objects

Based on the study results, we opted for the icon as the default representation. The reason behind this is that the icon has been shown to be acceptable and provides a sufficient level of detail about occluded items while introducing an acceptable amount of visual clutter to the workspace. Based on the participants’ feedback, for Projec^Top, we extend the icon representation from the study by adding an abstract outline visualization showing the outline of an occluded

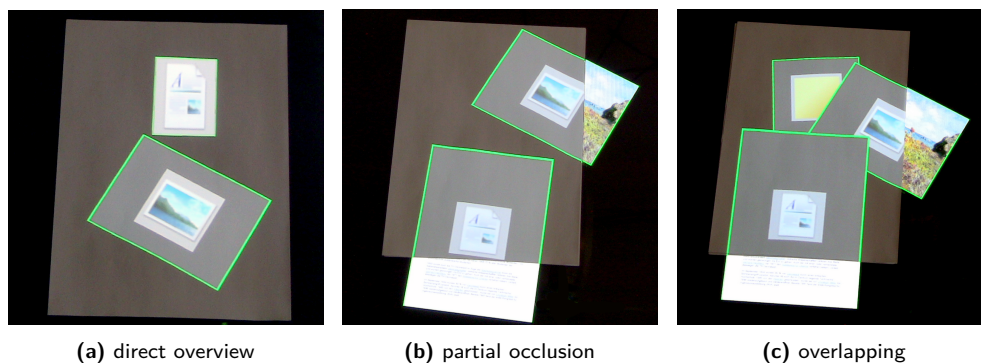


Figure 3.4: Visualization overview.

object (see figure 3.4). As a result, type, orientation, position, and size are perceivable by the user. Since the outlines overlap just as the occluded documents do, the user can roughly understand the layering of the occluded document and better recognize structures such as small piles of digital documents. Further, the chosen combination of an icon visualization with the outline allows the user to more efficiently distinguish between objects of the same type: For instance, when searching for a long e-mail or large image, the user can identify e-mails or images based on the icon and their length (or size) based on the outline. Thereby, the user can better estimate whether a relevant document is under a physical object without having to take any physical action (see figure 3.4a). Also, previous research [Khalilbeigi et al., 2013] shows that users tend to prefer moving lightweight objects instead of using the provided system support for occlusion resolution. The spatial awareness (position, size, and z-order are visible) provided by this visualization gives them a hint about where the occluded objects are, allowing a more directed and efficient movement of the occluder. Another argument for using the outline element in the visualization is that it can be used to “complete” objects in partial occlusion situations (e.g., only half of the object is underneath a physical occluder) as shown in figure 3.4b.

The ProjectTop system starts to visualize occlusion when a certain part (10%) of a digital object is underneath a physical object. This threshold ensures that digital objects that have, for instance, only the tip of a corner hidden under a physical object, do not cause disturbing projection artifacts.

Physical objects do not always occlude digital content fully, but only partially. This might even be a wanted case of occlusion, as only a part of the occluded object is actually needed, e.g., a small figure on a large page. In such situations, the ProjectTop approach has several advantages over tabletop-based approaches

[Furumi et al., 2012; Javed et al., 2011; Khalilbeigi et al., 2013] that use proxy objects like halos or icons on the tabletop display. These approaches require definition of a point where the transition between showing the actual object and showing the icon representation happens (e.g., if a digital object is occluded by 50%, the icon is displayed instead of the object). However, it is not clear when the user actually wants this switch to happen: For instance, in the case of a document with a relevant diagram on the lower third of the page, the user might accept that the rest of the page is occluded by an object since he does not need it. Hiding the occluded object in this case and displaying only an icon would be annoying to the user. On the other hand, for unimportant documents, an earlier transition might be desired (e.g., if 20% is occluded). However, it is difficult to automatically infer at what point the transition is acceptable. Additionally, there is a notable visual break during the switch when the large object changes to a tiny icon. In contrast, ProjecTop allows a continuous transition between full occlusion and full visibility without any abrupt, distracting change in the visualization as both styles can coexist in a meaningful way: The outline can complement the still visible part.

When multiple overlapping objects are occluded, the visualization chosen for ProjecTop naturally conveys their z-order, as the outlines occlude each other just like the represented objects do. The aforementioned smooth transition helps to identify the z-order even for two partially occluded items if the overlap happens under a physical object as shown in figure 3.4c.

The visualization is not only a static means to provide awareness about hidden objects, but also an interactive means to enable occlusion resolution without having to interact with the physical occluder. Besides retrieving the digital object from underneath the physical one, ProjecTop provides a set of multitouch gestures to interact with the occluded digital objects. These techniques are explained in the following section.

3.4.2 Interacting with Occluded Objects

In order to provide a consistent interface, ProjecTop supports standard object manipulation (moving, rotating, etc.) not only on the display of the tabletop but also on the surface of physical objects. This allows, for instance, natural manipulation of an occluded object “through” the occluding object.

In line with standard interaction on digital tabletops, a short single-finger tap gesture allows the user to change the z-order of digital objects by making

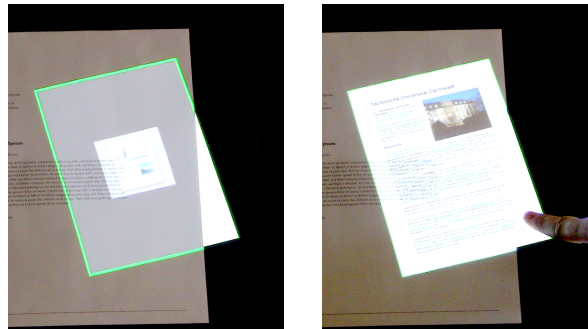


Figure 3.5: The Hold-to-Peek gesture in action.

the tapped one the foremost. This gesture is also supported on the tabletop surface as well as on the surface of physical objects. To support immediate access to and easy organization of digital objects in hybrid settings, we designed three techniques, which we describe in the following, and that are summarized in table 3.5.

Hold-to-Peek

Hold-to-Peek is a quick way of getting a glimpse of occluded objects. When selecting and holding an occluded object with one finger for a short amount of time, the occluding physical object becomes “transparent” by filling the outline representation with the actual content of the underlying object. This allows users to see through the occluder and look at the digital item. When the finger is lifted, the object returns to its default outline representation (see figure 3.5 for an example). This technique can be further extended by replacing the purely visual fade-in with a “semantic” fading, in which gradually more information is revealed as the user holds longer.

This approach has several advantages over prior approaches that are based on pulling the icons out from underneath the occluder: First, tapping and holding is a simple gesture that does not require any further movement. Second, the gesture can be performed on either the occluder or the tabletop, which mitigates the problem of the tabletop surface being inaccessible (e.g., between two tall objects). Because it shows the occluded object in place, it also requires no additional free display space next to the occluder, which is beneficial in very cluttered settings.

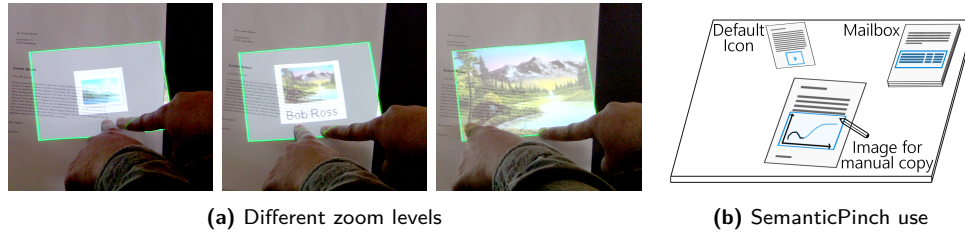


Figure 3.6: An overview of the SemanticPinch gesture.

SemanticPinch

The SemanticPinch provides permanent access to higher levels of detail about the occluded object. Similar to the typical pinch-to-zoom gesture, SemanticPinch is used to change the representation of an occluded digital object on a physical one (see figure 3.6a): As the user “zooms in”, more and more detail about the object is revealed. Contrary to the state-of-the-art semantic zooming facilities for occlusion [Khalilbeigi et al., 2013], the visualization remains in its current semantic zoom representation when the user lifts the fingers. This way, the user can control the amount of information he wants to receive about an object *permanently*.

Using this technique, users can maintain a more detailed view of occluded objects. Practical use cases are, for example, an occluded mailbox that the user can semantically zoom in on in order to view the last n received emails instead of just a mailbox icon. Due to the top-projection, the user gets an always-on-top mailbox overview. Another use case would be tracing digital objects to copy them on physical paper. The user can place the digital object, occlude it with a sheet of paper, and then semantically zoom in until the full object is shown, allowing him to trace it using a pen. See figure 3.6b for an illustration.

HyperMove

Moving objects is a key action for workspace organization and well supported on digital tabletops. With the addition of physical objects, moving digital objects becomes cumbersome: The direct moving path might be obstructed by a physical object, requiring the user to either move the physical object or drag the digital one around the physical one. In order to better support these cases, we developed HyperMove.

HyperMove is an extension of the moving gesture supported on conventional

Name	Purpose	Description
Hold-to-Peek	Peeking at an occluded document	Selecting an occluded object to get a temporary view of an occluded document by making the occluder visually transparent.
SemanticPinch	Controlling the amount of information	Permanently changing the level of detail of a representation of an occluded object gradually between an icon and more information (e.g., title or headlines) by performing a pinch gesture.
HyperMove (manual)	Moving digital documents	Single-handed gesture for moving a document (from) underneath or across a physical occluder, especially for shorter distances.
HyperMove (direct)	Moving digital documents	Bimanual gesture for moving a document (from) underneath or across a physical occluder, especially for longer distances.

Table 3.5: Summary of the interaction techniques included in the ProjectTop interface.

multitouch surfaces and is not bound to the display of the tabletop, but extends the interaction space onto the surface of physical objects. ProjectTop supports two variants of HyperMove: *manual* and *direct*.

HyperMove (manual) The manual version is an extension of the conventional single-finger dragging gesture known from the tabletop surface. Since touch input is available on physical objects as well, the gesture can now be used to drag digital content across a physical object without the need to drag it around. Since for physical objects thicker than a few pages a notable physical edge exists, the user needs to lift the finger when reaching the object boundary (i.e., drag until the user reaches the edge of the object, lift the finger, and continue the dragging on the surface of the object). For thin objects, the drag can be done without explicit lifting. The gesture is active as long as the finger is held down.

HyperMove (direct) The direct version uses a bimanual gesture to move digital items over large distances without having to drag them. The user references the desired position of the object by a static tripod-gesture (similar to the teleport gesture in [Khalilbeigi et al., 2013]) with one hand. Then, he can

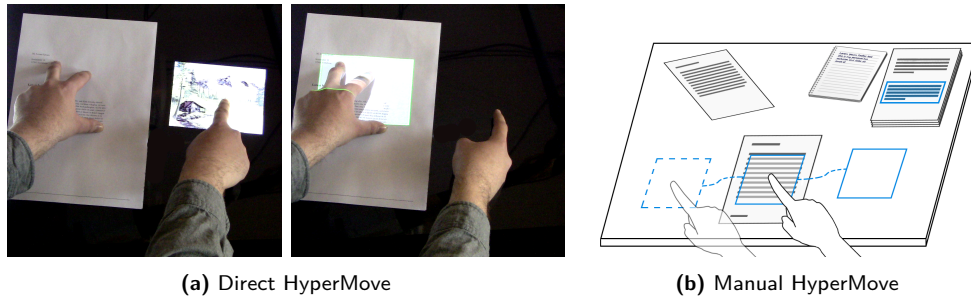


Figure 3.7: An overview of the HyperMove technique.

move digital objects immediately to this location by performing a single-finger flick gesture with the other hand on them.

Again, ProjectTop allows this gesture to be performed on the tabletop as well as physical objects (see figure 3.7a and 3.7b). Besides supporting conventional moving tasks, this gesture allows for more advanced and quick organization, like piling several digital objects under a set of physical ones.

3.5 User Study

In order to evaluate the interaction concepts presented in the previous section, we conducted a user study. The overarching goal of the study was to find out how the concepts provided by ProjectTop, i.e., the in-place resolution and the visibility of the spatial layout, perform under different clutter levels. Performance in this context is defined as the time that users need to find an occluded object and the effort required to do so (e.g., the number of interactions with physical objects).

<i>Method:</i>	Quantitative
<i>Participants:</i>	17
<i>Duration:</i>	ca. 60 min
<i>Data Gathering:</i>	Logging, video recording, questionnaire and semi-structured interview
<i>Variables</i>	
<i>Independent:</i>	Interface type, clutter level
<i>Dependent:</i>	Time, object interactions, errors made

Table 3.6: Study design overview.

Based on prior studies [Javed et al., 2011; Khalilbeigi et al., 2013], we decided to compare the top-projection-based ProjectTop approach with a system named “Baseline”, that relies purely on the display of the tabletop. Baseline consists

of two well-established occlusion support techniques: occluded objects are visualized as icons around the occluder to provide accessibility and a glowing visualization surrounding the occluding object provides general awareness [Javed et al., 2011]. This way, we can ensure generalizability and comparability of the results of our study. Additionally, we did not impose any restrictions on interaction with the physical objects (e.g., physical objects must not be moved) in order to force usage of the digital occlusion resolution techniques. It was left up to the user to decide whether or not to interact with the physical occluder. Thus, users were allowed to lift or move physical objects if they preferred to do so rather than using the provided system support in both systems throughout the study.

As we know from previous work, the usage and performance of occlusion-related techniques is dependent on the amount of clutter on the tabletop [Khalilbeigi et al., 2013]. To see whether this has any impact in a top-projection scenario, we compared ProjectTop with Baseline in two different settings: *low clutter* and *high clutter*. We rank the clutter level of hybrid tabletop settings as a function of the number of digital objects on the tabletop and the amount of occluded display surface. In this experiment, for the low clutter conditions, we used 10 digital objects (of which at least 5 were initially occluded) and 3 physical objects. This choice led to about 40% occluded display area. The high clutter condition consisted of 15 digital objects (10 were occluded) and 5 physical objects, leading to about 60% occlusion.

3.5.1 Apparatus and Material

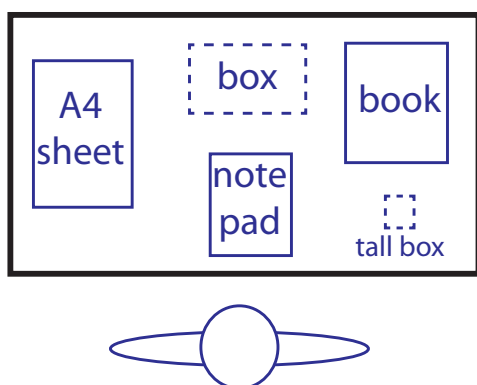


Figure 3.8: Initial setup. The dashed objects were only present in high clutter conditions.

We conducted the study using a C#- and DirectX-based implementation of the proposed techniques that runs on top of the PeriTop hardware setup described in chapter 2. Following previous studies [Khalilbeigi et al., 2013], we used the following items as physical objects: a thick book (20×24 cm with 4 cm height), a standard DIN A5 paper notepad, a printed A4 document, a cardboard box sized approximately A5 with a height of 3 cm, and a heavy, small elongated box ($7 \times$

7 cm with 14 cm height). Since the perception of projection is dependent on the texture of the projection surface, we printed the document with a lighter texture on it, and the book was chosen to have a darker one. The small box and the cardboard box were only present in the high clutter conditions. The other objects were present in both high and low clutter conditions.

We chose the objects to be widely used in an office environment with respect to size and ease of moving them. As the size of the tabletop is limited and the display could be damaged when high loads are placed on it, we had to abstain from other common objects such as monitors or a full keyboard. The initial physical layout for the study can be seen in figure 3.8.

As for digital objects, we chose a set of photos, documents, and virtual Post-it notes. Each of these types consisted of documents from several different categories; e.g., for the documents, there were documents about persons, movies, universities, etc. The objects were randomly chosen from the pool and placed randomly across the tabletop surface at the beginning of each trial.

3.5.2 Task

The experiment consisted of a search task in which the participants had to find and count a set of digital target objects, for example, the number of documents about movies, within a larger collection of various documents. The targets were mixed with other digital objects of different types or topics (distractors). All target objects were initially hidden under physical objects.

3.5.3 Hypotheses

we expected that, compared to Baseline, ProjectTop would reduce the time needed to find a specific object because in most situations, the occlusion can be directly resolved without having to perform a dragging gesture (e.g., to perform a semantic zoom on a proxy, which gradually reveals more information as it is pulled away from the occluder) but rather a single tap-and-hold. Due to the in-place occlusion resolution and the awareness of the spatial layout, we expected the users to make fewer errors. Additionally, interaction with physical objects should be reduced as ProjectTop provides an easy way to see where an occluded object is placed before lifting or moving the physical occluder. Therefore, these actions can be carried out more efficiently. In sum, the hypotheses to be examined were:

- H1: ProjectTop requires less time to find and access objects
- H2: ProjectTop reduces the number of errors users make
- H3: ProjectTop reduces the number of interactions with physical objects, and the distance they are moved

3.5.4 Experimental Design

The experiment was designed as a two interface (Baseline and ProjectTop) \times two clutter levels (Low and High) within-subjects experiment. The presentation order for the four conditions was counterbalanced. For each condition, there were five repetitions with different types (documents about movies, images of persons, etc.) and numbers (one to three) of targets. For each condition, the order of the target types for the five trials was randomly assigned for each participant. Prior to each set of trials, the users received a short introduction to the current interface and task and had some time to play around with the system until they felt comfortable using it. The users were then asked to perform the counting task as accurately and quickly as possible.

The primary measurement for the task was the completion time, which was measured by pressing the space bar of a keyboard within the participants' arm reach when starting and after finishing the counting. The number of interactions with physical objects as well as their movement distance was logged. For the digital objects, the number and type of special interactions (e.g., pulling an occluded object out from underneath an occluder) was logged. The number of targets reported by the participants was recorded and later compared to the actual number of targets present to determine the participants' error rate. The participants advanced to the next trial without being told whether they were right in order to avoid influencing their behavior (e.g., trying to be faster or slower).

After each condition, the participants were asked to fill in a NASA-TLX [NASA Ames Research Center, Moffet Field, 1988] questionnaire to measure the perceived workload. At the end of each session, the participants were interviewed in a semi-structured way to gather additional subjective feedback. We placed emphasis on the appropriateness of the projection during the interview. The entire sessions were recorded on video and lasted about one hour.

3.5.5 Participants

We recruited 17 participants (15 male, 2 female, all but 2 right-handed) aged between 21 and 46 ($\mu = 27.588$, $\sigma = 5.657$) from our university (students and employees). There was no compensation provided. All participants had previous experience with touch interfaces, and all except three participants had previous experience with interactive tabletops.

There was a total of $17 \text{ participants} \times 2 \text{ interfaces} \times 2 \text{ clutter levels} \times 5 \text{ repetitions} = 340 \text{ trials}$.

3.6 Results and Discussion

After preprocessing of the recorded trial data (e.g., removing outliers more than three standard deviations from the mean), nine trials had to be removed due to errors in time measurement. To analyze the data of the remaining 331 trials, we used a two-way repeated measures ANOVA (ANalysis Of VAriance). An initial analysis of the recorded data confirmed that there was no significant influence of the presentation order of the tasks for any measure, e.g., any learning effect on the interface. Therefore, the presentation order was neglected in the following analysis. For all pair-wise comparisons, Bonferroni correction was applied where necessary. The Shapiro-Wilk test was used to check for normal distribution and ensure the applicability of ANOVA.

3.6.1 Errors

When looking at the average number of wrong answers given by the participants, we observed a significant ($F_{1,16} = 8.209$, $p = 0.011$) difference between ProjecTop and Baseline in favor of ProjecTop (see figure 3.10 right). Using ProjecTop, users made 0.324 (SE = 0.085) errors on average, while in the Baseline condition they made more than twice as many, with an average of 0.735 (SE = 0.143). This supports H2. The influence of the clutter level was also significant ($F_{1,16} = 4.68$, $p = 0.046$), showing a higher number of errors for the high clutter conditions ($\mu = 0.676$, SE = 0.121) than the low clutter conditions ($\mu = 0.382$, SE = 0.110), as one would expect (see figure 3.10 left). There was no significant effect of interface \times clutter on the number of errors.

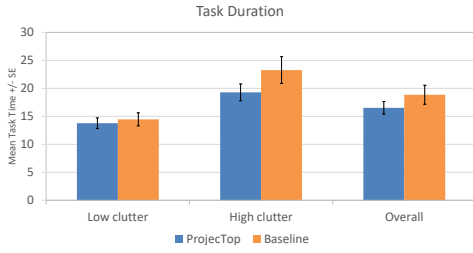


Figure 3.9: Average task duration by interface.

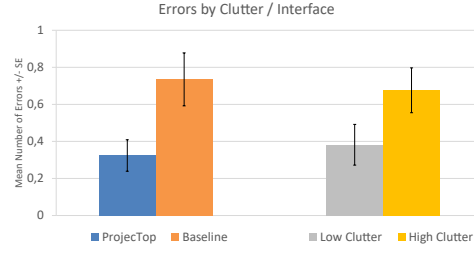


Figure 3.10: Average number of counting errors per trial by clutter level (left) and by interface (right).

3.6.2 Task Duration

The analysis of the measured trial time revealed that the interface as well as the clutter level had significant influence. ProjectTop ($\mu = 16.5$ s, $SE = 1.7$) performed significantly ($F_{1,16} = 6.09$, $p = 0.025$) faster than Baseline ($\mu = 18.9$ s, $SE = 1.1$), see figure 3.9. The clutter level also had a significant ($F_{1,16} = 34.52$, $p < 0.001$) influence on the trial time ($\mu = 14.1$ s, $SE = 1$ for low clutter; $\mu = 21.3$ s, $SE = 1.9$ for high clutter). Additionally, there was a significant effect of interface \times clutter on the completion time ($F_{1,16} = 6.67$, $p = 0.02$). Pairwise comparison revealed that ProjectTop performed significantly faster under the high clutter condition ($p = 0.017$). For the low-clutter condition, the difference was not significant ($p = 0.29$).

These results show that ProjectTop performs faster than Baseline, especially in highly cluttered environments. This supports H1. Interestingly, despite being faster, the participants made significantly fewer mistakes (reporting a wrong target count) using ProjectTop than they did with Baseline as described above. One reason for this may be that with Baseline, icons that were shadowed by a thick or high object were not visible from the users' point of view. This correlates with a higher probability that those objects were missed (leading to more errors) or that users needed to take special care of them by explicitly looking behind objects (leading to additional time). ProjectTop inherently circumvents this problem as the visualization is right on the occluder rather than behind it.

3.6.3 Interactions with Objects

When analyzing the number of interactions with physical objects, we found that the interface had a significant influence on the number of interactions

with physical objects ($F_{1,16} = 9.668$, $p = 0.007$) and the distance physical objects were moved ($F_{1,16} = 5.72$, $p = 0.029$). In ProjectTop conditions, the number of interactions was slightly higher with an average of 1.38 interactions per object, whereas Baseline had an average of 1.07. The same holds for the distance objects were moved: Baseline conditions had an average distance of 62.99 mm per object while ProjectTop conditions had an average of 96.55 mm. Hence, H3 is rejected. There was no significant effect of the clutter level or interface \times clutter.

Even though the amount and distance objects were moved was significantly higher with ProjectTop, the absolute difference between the two interfaces is rather small at about 3 cm. Interestingly, interaction using the ProjectTop concept was, despite the larger movement distances, less time-consuming and users made fewer errors. A possible explanation was the spatial awareness provided by the icon plus outline visualization. It gives users a strong visual clue regarding the spatial layout of the occluded digital objects before they begin interacting with (e.g., moving or lifting) the physical occluder. As a result, no additional time is needed to interpret the layout underneath the occluder. This explanation is supported by analyzing the video recordings, which confirm that users tend to move the occluding object away more often with ProjectTop, since they can do so with a single directed movement.

When analyzing the number of pull-out interactions with digital objects (i.e., the digital object is pulled completely out from underneath the occluder), we found no significant difference between ProjectTop and Baseline ($F_{1,16} = 0.73$, $p = 0.407$). Although the measured number was slightly higher with ProjectTop (11.62 for ProjectTop and 10.62 in the Baseline condition), we conclude that the interface has no influence on the number of interactions with digital objects.

3.6.4 Task Load Index

The analysis of the NASA TLX questionnaire revealed that the interface had no significant influence on the measured categories (see figure 3.11 for an overview). There was also no significant effect of interface \times clutter except for the perceived effort ($F_{1,16} = 4.848$, $p = 0.043$): Baseline has values of 9.82 for the high clutter condition and 6.18 for low clutter ($p = 0.011$). In contrast, ProjectTop has 7.71 for high and 7.12 for low clutter respectively ($p = 0.365$). The clutter level had significant influence on all measures except performance ($p = 0.17$), with mental demand ($p = 0.005$), physical demand ($p = 0.037$),

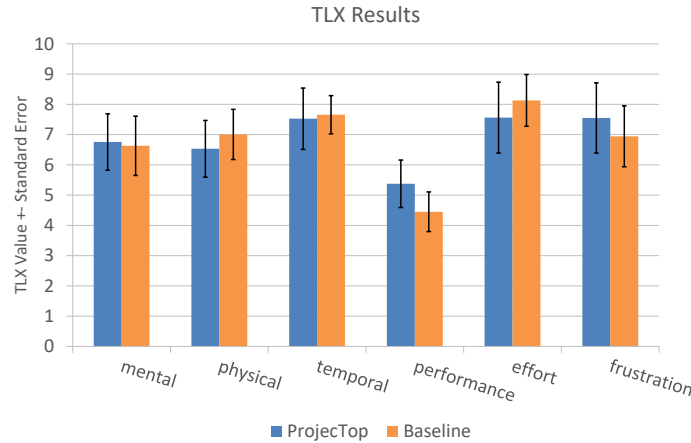


Figure 3.11: NASA TLX results (the full scale ranges from 0 to 20).

temporal demand ($p = 0.012$), effort ($p = 0.01$), and frustration ($p = 0.055$) being lower for the low clutter conditions than for the high clutter conditions, as one would expect.

3.6.5 User Feedback

The participants’ feedback in the post-interviews regarding top-projection varied: Three users found the ProjectTop visualization to be “*chaotic*” (P11) compared to the baseline. A few participants explicitly criticized the interference between projected digital and physical text on the objects, which both become unreadable if they overlap. However, the majority (12 out of 17) participants liked top-projection, for example, “*ProjectTop’s projection on objects is very cool*” (P10). Some participants even wanted the full content of digital objects to always be projected without user interaction (P10, P15, P16, P17), instead of the icons. P12 found ProjectTop to be “*clearer [than baseline] with many objects*”, and others stated that they had difficulty obtaining an overview in the Baseline condition (e.g., P15: “*Did I already take a look at that icon or not?*”).

Top-projection A common statement was that top-projected visualization should only be used for the storage area, but not for the working area (e.g., P12: “*Projection is only good in storage areas and not for the working area, because one concentrates there on specific objects and doesn’t want to overlook the mass of objects*”). At the same time, the concern was raised that the question of where to draw the border between the working and storage areas is difficult to answer. To resolve this problem, users suggested a differentiation by focus so

that physical objects the user interacts with do not get top-projection, while the others do (P2, P10). An alternative suggestion was to vary the brightness of top-projection instead of deactivating it, so that for objects in focus, the projection becomes darker to improve readability of the physical content (P15). P16 suggested a combined Baseline/ProjecTop system in which the system decides whether objects are suitable for top-projection (using ProjecTop) or not (using Baseline style). The user should then, however, be able to switch between the styles for specific objects manually. In the context of a combined system, P2 wanted to have the Hold-to-Peek gesture for the icons in Baseline, so the actual object is top-projected when touching its icon. Also, a combination was suggested in which a Baseline-style icon visualization would be used on free spots on the documents (i.e., the icons are moved to a close-by free area on the occluding object), thereby reducing interference between the content of the document and the projection.

Accessing occluded objects All users found accessing an occluded object through ProjecTop easy. The Hold-to-Peek gesture was even found to be sufficient to complete the task by some users, e.g., “*ProjecTop was often sufficient to see what’s underneath*” (P6). However, problems occurred in the case of digital/digital occlusion, which is currently not resolved by ProjecTop. In the case of one digital object being fully occluded by another under a physical object, the occlusion was reflected by the visualization, making the bottom digital object inaccessible. The icon representation used

by Baseline inherently resolves this occlusion, as the icons do not overlap. This suggests the addition of a similar mechanism to the ProjecTop concept (e.g., a combination of the Baseline and ProjecTop concepts), which causes no further scientific challenge. Additionally, P14 suggested adding different outline colors to group objects into categories and further speed-up finding objects as shown in figure 3.12.



Figure 3.12: Differently colored outlines to categorize objects.

Spatial awareness In general, the participants found the concept of spatial awareness employed in ProjecTop useful, especially if they wanted to lift the occluder to access one or more occluded objects: “*Spatial awareness was beneficial. I could look at the physical object and directly saw where the objects are. With Baseline I had to look around the [physical] object to see where*

[digital] objects are” (P15). Besides the concerns regarding interference, there were no negative comments on this topic.

Touch on objects Two users found the touch function on arbitrary physical objects strange: P4 found “*touching flat objects more intuitive than thick objects, as flat ones are closer to the screen, which is naturally interacted with using touch. For thick or uneven objects, it was strange to touch the surface in order to interact.*” P1 said that, “*for thick objects, there is a problem if a gesture leaves the object, for instance when dragging out an object from underneath a book, as the touch surface is not continuous at the book’s edge.*”. The other participants were positive about this functionality, so it can be assumed to be an appropriate way of interacting with top-projected content, at least for a significant share of users.

3.7 Implications

Based on the study results, we propose three main implications for informing the design of future hybrid tabletops augmented with top-projection:

How to resolve occlusion? Top-projection and in-situ or on the tabletop display and off-place? The study showed that, in general, top-projection is a suitable means to extend the display and interaction spaces in hybrid settings and particularly to resolve physical/digital occlusion. However, its use must be considered carefully to provide support to the user without distraction or hinderance. As a result, it cannot be applied to every object on the workspace in the same manner. In order to avoid conflicts with the user’s task and needs, whether to use top-projection or not must be decided on a per-object basis. This decision is influenced by two main factors: The user’s current task and preferences on one side and the properties of the occluder on the other.

The first factor is driven by the user’s needs during a task, e.g., top-projection on a document the user is currently reading is clearly unwanted and, as P12 suggested, should be switched off in the working area. In order to do so, the working area of the user must be known. A trivial approach would be to define a static area on the desk as the working area. However, this would not take into account the dynamics of knowledge work processes. By monitoring the activities of the user, e.g., movement of documents, interaction with documents, and the user’s position, the current focus area of the user could be inferred

dynamically. Based on this information, the visualization style can be adjusted in real time so that within the user's focus area, top projection is not used in favor of tabletop display-based visualizations. With increasing distance to the focus zone, top-projection will be enabled and its brightness increased while showing greater and greater detail.

The second factor is objects that are not suitable for projection or interaction. This can be either because they are very dark or strongly textured, making the projection unrecognizable, or because the objects use touch as input modality (e.g., keyboard or tablet), which would interfere with the user wanting to interact with the projection. Again, switching to a tabletop-based representation is a possible solution. The system can be enabled to recognize common unsuitable objects and switch automatically.

Besides these two factors, which can be considered programmatically, it is also sensible to provide the user a means of individually controlling top projection so that he can turn it on or off for specific objects at will. This is supported by the comments of the participants: Some wanted even more projection and some already felt distracted by the current visualization. Also, some users said they would prefer to have a gesture provided by the system to toggle top-projection globally. Then, they can use top-projection like an expose mode when they need an overview without being otherwise distracted.

Is spatial awareness useful? Despite the higher movement count of physical objects and a large number of interactions, ProjecTop was significantly faster in the high clutter condition. Video analysis confirmed that users often lifted all objects in Baseline and ProjecTop conditions prior to finishing the task in order to check for missed objects. Even if subconscious, the spatial awareness in ProjecTop likely facilitates this process because participants already knew the underlying layout, leading to faster completion times and fewer errors. It is therefore essential to provide such visual clues, even if users don't consciously use them and keep interacting with the physical objects to resolve occlusion.

How to deal with touch across tall objects? A problem with expanding the interaction space onto the physical objects is the discontinuity of the touch surface, as two participants mentioned: The steep edges of physical objects make smooth touch interactions impossible because the user has to lift his finger in a step-like manner to always maintain contact with the touch surface. A possible solution to this problem is a kind of *hover-touch*, which allows continuing the move while keeping the finger in the air. This enables a smoother transition between the surfaces, as the user can start lifting the finger before reaching

the edge of a physical object.

3.8 Summary and Conclusion

This chapter addressed one of the core issues identified in the motivation chapter of this thesis, namely the occlusion of digital content through physical objects. In order to foster better integration of physical and digital documents and facilitate the use of spatial layouts by users, this chapter presented the ProjectTop concept, an in-place physical/digital occlusion resolution approach.

ProjectTop is a first step in circumventing occlusion by using the surfaces of physical objects in addition to the tabletop surface. Thereby, the formerly foreign objects are made part of the interface by using them as additional interactive display surface to resolve the occlusion situation in situ. In order to ground the design of the visualization techniques used in ProjectTop, we conducted a first study comparing different visualization styles for use in the context of in-situ top-projection on documents.

In a second step, the ProjectTop concept was evaluated as a whole, revealing that the top-projection approach helps resolve occlusion-related problems like awareness and access and to be a valuable addition to interactive surfaces if used advisedly. However, the study indicated that top-projection has also to be seen critically since it causes visual clutter that is negatively perceived. As a result, it can be concluded, that the user's focus and task have to be considered when deciding whether to use top-projection for a specific object or not. Generally, the qualitative feedback indicated that the spatial awareness is perceived as useful by the participants.

In order to inform the design of future, improved in-situ occlusion resolution systems, a set of implications for the use of top-projection on physical objects derived from the results of the study was presented.

As a next step, the more complex scenario of hybrid stacking will be explored in the following chapter. The results of the studies presented in this chapter are used to improve the design of the top-projected visualization techniques used to visualize hybrid stacks.

StackTop – From Occlusion to Hybrid Stacking

This chapter proposes the “StackTop” concept, a set of interaction and visualization techniques that support ordered hybrid digital/physical piling (hybrid stacking) on interactive surfaces. StackTop enables a tighter physical/digital integration in hybrid workspaces and provides a more consistent approach when working with hybrid document sets.

The chapter begins with an introductory section (section 4.1), after which the current state of the art regarding piling is analyzed in section 4.2. In section 4.3, the StackTop concept with its interaction and visualization techniques is presented. The designed interaction techniques are then evaluated in a first user study that is presented in sections 4.4 and 4.5. Based on the results, a set of design implications derived from the study is presented to inform the further development of systems for hybrid stacking support in section 4.6. The chapter concludes with a summary in section 4.7.

4.1 Introduction

In the previous chapter, we addressed occlusion in general and provided a means of awareness for otherwise invisible digital objects. However, in that context, there was still a strict separation of digital and physical content assumed: The digital documents were confined to the table surface and the physical objects placed on top. While this was a first step towards the goal of tighter integration of physical and digital content, it still does not provide support for arbitrarily placing digital and physical documents on top of each other, as it is possible in the physical world with piles of paper documents or in the digital world with a set of all-digital documents.

In this chapter, we want to expand the piling metaphor to the domain of mixed physical/digital piles. As mentioned previously (section 2.3.3), piling

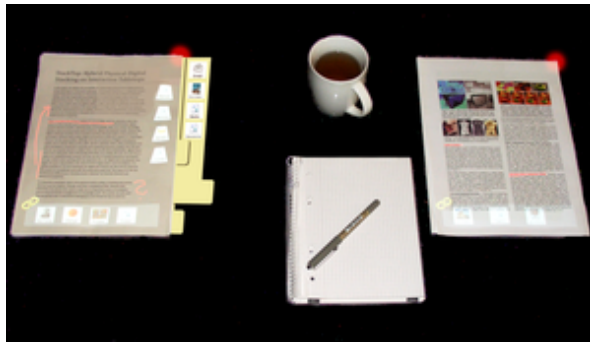


Figure 4.1: StackTop facilitates mixed digital/physical (hybrid) stacking on interactive tabletops by providing awareness and means of access to stacked documents.

can be seen as a special form of occlusion where the process of occlusion is meaningful, and the order of documents may be relevant. While this also holds in purely physical and purely digital settings, in this chapter we focus particularly on the concurrent use of digital and physical documents. In order to underline the aspect of ordered piling and to distinguish it from the (unordered) pile defined by Malone [1983], we refer to such piles as (*hybrid*) *stacks* from now on.

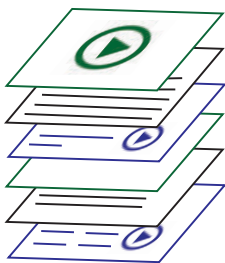


Figure 4.2: Schematic of a hybrid stack: digital (green), hybrid (blue), and physical (black) documents can be stacked arbitrarily.

While it is, with current tabletop systems, already possible to arbitrarily stack a set of purely digital or physical documents, true hybrid physical-digital stacking remains a salient challenge. A number of prior works [Fujii et al., 2003; Iwai and Sato, 2006; Khalilbeigi et al., 2010; Wellner, 1993] have started to address the challenges of hybrid media usage; however, the developed techniques and systems mostly use the two-dimensional display space provided by the interactive surface (e.g., tabletop display). The digital content is thereby inherently bound to the two-dimensional display surface. True hybrid stacking, on the other hand, requires a three-dimensional projection space with multiple layers, which those works did not consider.

Extending beyond previous work, this chapter contributes StackTop, an integrated set of visualization and interaction techniques for *true hybrid stacking*. A hybrid stack in this context is an *ordered* pile consisting of both, digital and physical documents (see figure 4.2). Unlike hybrid piles [Khalilbeigi et al., 2010], where the physical and digital parts are separated, StackTop allows

seamless interweaving of both types of documents in one stack and supports users in navigating, searching, and reorganizing both document types within the stack.

4.2 Related Work

The work presented in this chapter closely relates to the field of augmented desktops in general, and occlusion and digital support for stacking in particular. We will discuss the relevant body of related work in the following.

4.2.1 Augmented Desktops

Augmented desktop systems form the basis for our contributions in this chapter because in order to be able to support hybrid stacks, pertinent means of in- and output as well as ways of identifying the documents are required. As in the previous chapter, we omit the detailed discussion of augmented desktops here to avoid redundancy. For details, refer to section 2.2.1.

4.2.2 Occlusion

Obviously, stacking of documents leads to occlusion. As a result, the related work on occlusion presented in section 3.2 is also relevant for the contributions in this chapter. However, in this chapter, we go beyond basic occlusion resolution by extending the possible locations of digital content to anywhere within the stack. Further, StackTop provides means of access to the whole stack with its digital and physical content instead of just for single digital or physical documents only.

4.2.3 Stacking Support

Support for stacking (or piling) of documents is only coarsely covered in related work. While it is well explored in purely physical environments and as a metaphor in purely digital settings, and while some studies were conducted on the impact of hybrid piling, there is no work that supports fully interweaved digital/physical stacking yet. The relevant related work in this field is discussed in the following.

Stacking Support for Physical Environments

Purely physical stacks are very common in workplaces today. As a result, techniques have been developed to support interaction with those stacks digitally. Fujii et al. [2003] developed a tangible search concept that tracks the objects in a physical stack so the user can move his finger along the side of the stack and view thumbnails of the objects contained in the stack. In a similar direction, Iwai and Sato [2006] developed the limpid desk. It resolves occlusion between multiple piled physical documents by projecting the content of the physical document touched by the user on top of the pile. In both cases, the user is provided with simple means (sliding along the side of a stack or tapping a document) to gain access to items within the stack without having to manipulate the stack itself. This is a valuable feature when the user simply wants to get a quick glance at a single object. Therefore, it should also be adapted in the context of StackTop in a way that allows using it also for digital documents within a stack that might not be directly accessible.

Additionally, support for stackable tangibles has been explored by previous research: the CapStones by Chan et al. [2012] are tangibles that can not only be used directly on the tabletop surface, but can also be stacked on top of each other. By creating specific touch-point patterns on a capacitive touchscreen, the system can then detect the order and type of tangibles stacked, thereby creating a digital model of the physical stack. In order to create the patterns, the tangibles need to be specially crafted, and the number of stacked tangibles is limited by the number of touch points supported by the touch sensor and its resolution. Theoretically, it would be possible to apply this concept to stacks of documents as well, e.g., by using conductive ink to add the required electrodes to the paper documents. However, this would require modifying existing documents and due to the limitations mentioned above (number of recognizable touch points and resolution), it is likely impractical for larger collections of documents.

Stacking Support for Digital Environments

On the purely digital side, stacking is also very prominent: Mander et al. [1992] as well as Agarawala and Balakrishnan [2006] developed and employed a piling metaphor to facilitate digital document organization. While it is also possible to somehow “pile” icons in traditional desktop systems by placing them close together, they provide more explicit means for piling support: Agarawala et

al.'s BumpTop, for instance, integrates a lasso tool that allows the user to group a set of documents selected by the lasso and order them in a tidily aligned pile. In the other direction, BumpTop provides the ability to expand a tidy pile into several spread-out arrangements, e.g., a grid or fan-out. This allows users to easily gain an overview of the contents of a pile. In the physical world, piles are not always tidily aligned with precisely arranged documents on top of each other. Thereby, it becomes possible for the user to add more structure to the pile, e.g., by controlling the occlusion of documents within the pile, making some more visible than others. In order to leverage this aspect, piling concepts were developed [Bauer et al., 2004; Scott et al., 2005] that support more spread-out piles where documents do not need to be directly on top of each other despite being still recognized as a group. However, this comes at the cost of a reduced overview and larger space requirements. Along this line, Watanabe et al. [2007] developed support techniques for the organization of piles and similar spatial aggregations by automatically recognizing spatial groups of objects and visualizing them using a bubble-shaped outline. Based on this, the system supports operations like moving or copying the whole group directly. In the other direction, the system allows the user to spread out grouped documents without overlap to facilitate selection of individual objects from the group.

Jakobsen and Hornbæk [2010] explored different strategies to work with overlapping or tabbed documents on a screen. Piling was shown to perform well in retrieving the relevant documents, underlining the importance of piling. While techniques like spreading out or automatically tidying up a pile to provide an easy and direct overview of its contents are inspiring, these systems are all limited to the digital domain and the presented techniques do not work in hybrid settings. This is mainly due to the reason that it is not easily possible to relocate physical documents via software.

Stacking Support for Hybrid Environments

Steimle et al. [2010b] conducted an extensive user study to understand how natural practices of working with piles transfer to hybrid digital-physical scenarios on interactive surfaces. They found that concepts prevalent in the physical world, like piling, grouping, or sorting, are also applied to hybrid settings. They observed that participants frequently build hybrid piles by placing digital items first, then placing physical documents on top. Building upon this, Khalilbeigi et al. [2010] proposed a set of techniques to support

hybrid piling on tabletops. In these two concepts, construction of hybrid piles happens implicitly without the need for performing a special “hybrid piling gesture” or the requirement for specially crafted documents (e.g., equipped with electrodes, etc.). In this way, interaction with a hybrid pile is similar to interaction with a conventional physical pile. While inspiring, these approaches only support hybrid piles that consist of two distinct digital and physical parts.

Physical stacking of digital content by means of projected displays on sheets of paper [Holman et al., 2005] has been explored. In this case, the physical paper sheets serve as a tangible representation of single windows in a conventional desktop operating system. In addition to binding windows to paper sheets, interaction techniques for stacking, combining, or annotating are provided. Later, the use of eInk displays for document organization and access was explored by Girouard et al. [2012]. They developed the DisplayStacks system, which enables users to stack digital documents shown on flexible eInk displays just as they would with physical printed copies. The system then tracks the arrangement of the eInk displays, and this serves as a basis to provide – for instance – contextual overviews and other access techniques. The DisplayStacks are, however, limited to stacks containing only augmented eInk displays. As a result, stacking with other physical documents is not directly possible. Further, the number of digital documents is somewhat limited by the number of available eInk displays. Also, in the foreseeable future, eInk displays will remain rather expensive compared to paper and still lack the mechanical flexibility paper provides. Hence, eInk-based solutions are currently impractical for larger deployments.

Requirements and Summary

In the following, we present a set of requirements for hybrid stacking support derived from reviewing the related work and use them to compare the related work (see table 3.1).

- R1 **Physical documents** Physical documents are still widely used and an important part of hybrid environments. Thus, systems aiming at supporting stacking in hybrid environments need to support them.
- R2 **Digital documents** Digital documents are the second important part of hybrid environments and hence need to be supported as well.
- R3 **Coupling styles** In the real world, different coupling styles exist, e.g.,

	Physical Documents	Digital Documents	Coupling Styles	Arbitrary Placement	Easy Access
<i>Physical Stacking</i>					
Fujii et al. [2003]	●	○	○	○	◐
Iwai and Sato [2006]	●	○	○	○	◐
Chan et al. [2012]	●	○	○	○	●
<i>Digital Stacking</i>					
Mander et al. [1992]	○	●	○	○	●
Agarawala and Balakrishnan [2006]	○	●	○	○	●
Bauer et al. [2004]	○	●	○	○	◐
Watanabe et al. [2007]	○	●	○	○	○
Scott et al. [2005]	○	●	○	○	◐
Jakobsen and Hornbæk [2010]	○	●	○	○	●
<i>Hybrid Stacking</i>					
Khalilbeigi et al. [2010]	●	●	○	○	◐
Holman et al. [2005]	◐	●	◐	◐	●
Girouard et al. [2012]	●	●	○	○	○

Table 4.1: Properties of selected stacking approaches. ● indicates that a requirement is fulfilled, ◐ indicates partial fulfillment, and ○ that a requirement is not fulfilled.

tight coupling by stapling two documents together or loose coupling by just placing them on top of each other. These fundamental styles should be supported.

R4 Arbitrary placement In order to form true hybrid stacks, placing digital and physical documents arbitrarily on top of each other must be supported.

R5 Easy access Hybrid stacks are more complex to use since they contain graspable physical as well as immaterial digital documents. Hence, techniques for easily browsing a hybrid stack as a whole and retrieving or adding documents need to be provided.

4.3 StackTop

The underlying idea of StackTop is to allow arbitrary stacking of digital and physical documents. The notion of our “hybrid stack” is related to the definitions of “pile” and “file” given by Malone [1983]: it is ordered (like the file), but not every document has a title (like in a pile).

In summary, a *hybrid stack* is an ordered set of digital and physical documents that are stacked on top of each other and that are tidily aligned.

Besides physical and digital documents, prior work has explored the use of digitally augmented physical paperwork, e.g., by allowing the user to add digital annotations to physical documents. Basically, these augmented documents also consist of digital content “stacked” on top of the physical document. The big difference to hybrid stacking is the *coupling* between the digital and the physical document. The digital augmentation is tightly coupled to the physical document – it always sticks to it. In order to denote this, such documents are referred to as “hybrid documents” in the following.

Unlike hybrid documents, the digital content in a hybrid stack is more loosely coupled: If the whole stack is moved, the digital documents move along with the stack, but if a physical document is removed from the stack, the digital content that was on top of it remains within the stack. Both coupling styles are provided by StackTop in one integrated system.

Generally speaking, standard multitouch gestures [Villamor et al., 2010], like moving, zooming, and rotating are supported throughout the system.

4.3.1 Hybrid Documents (Tight Coupling)

In order to support a broad spectrum of digital document augmentations, StackTop distinguishes two types of digital objects: Annotations and linked documents. The presented techniques are summarized in table 4.2.

Annotations

The first category, annotations, comprises, for instance, digital pen strokes or virtual Post-it notes. Just as with real pen strokes or physical Post-its, it is sensible to show them where the user actually drew/placed them in their full size (see figure 4.3a).

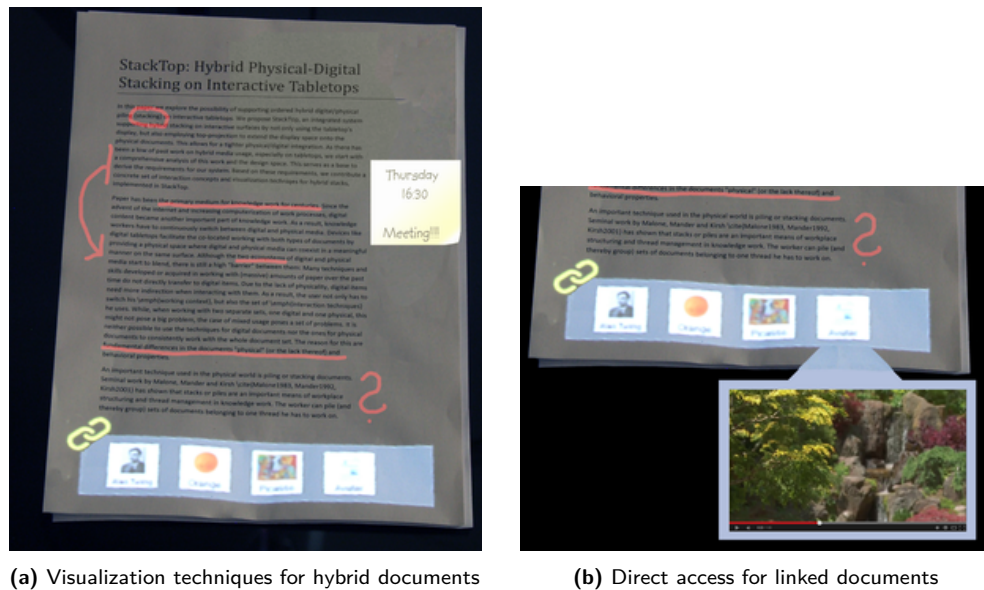


Figure 4.3: Tight coupling visualizations.

Annotations are created in different ways, depending on their type:

- *Pen-based annotations* are created by drawing with a digital pen directly on the document.
- *Annotation objects* like digital Post-its can be dragged from the tabletop surface directly onto the document, or from one document to another, and are automatically treated as annotation.
- *Other objects*, like images, can be used for annotation purposes as well. To do so, the user must first link the object as described in the next section, then move it from the link-bar to the final position on the document. By explicitly linking the object first, the coupling style (tight coupling in this case) is explicitly established and used to distinguish the action from the more frequent process of dragging a digital document onto the stack in order to insert it into the stack.

Linked Documents

For the linked documents, the exact location on the document is usually not relevant as these are associated with the whole document and not a specific part of it. They can therefore be visualized in a reduced form at free spots on the document in order to minimize visual clutter.

Name	Purpose	Description
Annotation	Digitally annotate a document	Add digital annotations to a document, e.g., free-hand drawings or virtual Post-its.
Linking	Link digital information to documents	Permanently link related digital documents to another document.
Content Access	Extract content from documents	Extract a part of a document, e.g., a figure, from a physical document.

Table 4.2: Summary of interaction techniques for hybrid documents included in the StackTop interface.

StackTop provides a “link bar” at the bottom of every digital or physical document where documents that should be linked can be dropped (see figure 4.3a). To view the linked documents, the user can either employ the remote peek gesture (for a quick, temporary view) [Khalilbeigi et al., 2013] or tap one of the icons to get a permanent view next to the physical document. As shown in figure 4.3b, the content is displayed in a “bubble” and can be viewed. To remove an item, the user can simply drag it from the link bar to the tabletop. In this way, linked documents are directly integrated with the physical document. Also, as a result, they do not consume additional screen space when not being accessed. This is unlike many existing systems that rely on the tabletop display to show linked documents [Everitt et al., 2008; Steimle et al., 2009].

Content Access

Further, StackTop provides digital access to the content of physical documents. When working with physical documents that are several pages long or with several single-page documents in parallel, one often needs only parts of other pages when reading or working with one page (for instance, the reference list, a referenced figure, etc.). StackTop allows the user to drag out sections or figures of physical documents by performing a tap-and-hold gesture on the heading of a section, paragraph or figure to select them. The selected part of the document is then highlighted, and a digital copy can be dragged off the document. The user can either have it on the tabletop for reference purposes or place it directly on another document as an annotation or linked document. This feature requires an annotated digital copy of the document to be available. The physical document is matched with the digital copy, which is then used to

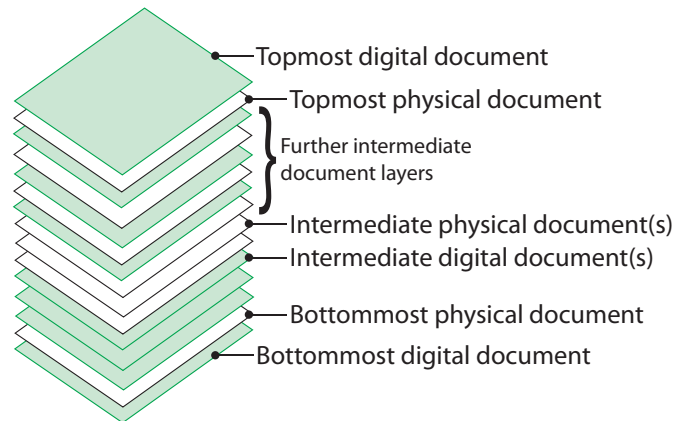


Figure 4.4: Nomenclature overview for the different layers of a hybrid stack.

retrieve the content.

4.3.2 Hybrid Stacks (Loose Coupling)

Hybrid stacks within StackTop may consist of digital and physical documents as well as hybrid documents as described in the previous section. StackTop indicates that a stack is a hybrid stack that is augmented by StackTop with a red glowing corner (stack indicator). In order to facilitate hybrid stacking, StackTop supports the following operations: building the stack, browsing and accessing its items or documents, and reordering the stack. For each of these tasks, StackTop provides corresponding techniques presented in the following sections. Figure 4.4 provides an overview of the nomenclature used in the following when referring to documents at specific positions within a stack.

Building the Stack

In the physical world, stacking and unstacking of documents are simple and intuitive operations: Documents are put on top of a stack, or by lifting a part of the stack, at a position within. To remove a document, one either takes the topmost document directly from the top of the stack, or manually lifts a part of the stack to access a document at an intermediate position within the stack. In order to find a document or the right place to insert one, the process of lifting parts may need to be repeated until the correct place is found.

To provide a similar interaction for hybrid stacking, StackTop rebuilds this behavior for digital documents: To add or remove digital documents, the user can either drag them directly on (or off) the topmost physical document of a

stack via a manual HyperMove or by using a direct HyperMove gesture (cf. section 3.4) to move them directly over a larger distance. In both cases, the digital documents are placed at the topmost position in the stack. In order to place a digital document at an intermediate level, one would, as with a physical stack, temporarily lift a part of the stack with one hand to insert a digital document at the intermediate position. As with physical stacks, the step of lifting documents might need to be repeated in order to reach the desired position within the stack. Similarly, removal of documents is possible by dragging a document from the stack or by HyperMoving it away. Physical documents are added and removed just as in a conventional purely physical stack.

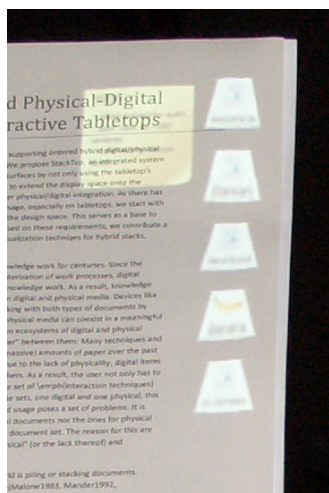


Figure 4.5: A physical document with a digital Post-it. The in-place access visualizes the digital documents above the physical as a stack of icons in the free margin of the document.

By placing the lifted part held in-hand back on the original stack after inserting or removing a document, the insertion process is completed. Consistent with the behavior in the physical world, placing the in-hand part on another empty area of the desk splits the stack into two stacks. Likewise, placing the lifted part on another stack adds that part to the other stack.

Currently, lifting of stack parts is bound to interaction with physical objects since they can be directly grasped. In order to insert a digital object at the correct position within multiple intermediate digital documents, the user can drag it to the appropriate position using the in-place access described in the next section. The in-place access is shown in figure 4.5. To minimize visual clutter, the digital documents are shown as icons.

Browsing and Accessing

A common task with stacks is to browse and access information within them. StackTop offers three techniques to support this: *in-place access*, *registers*, and *StackPath*.

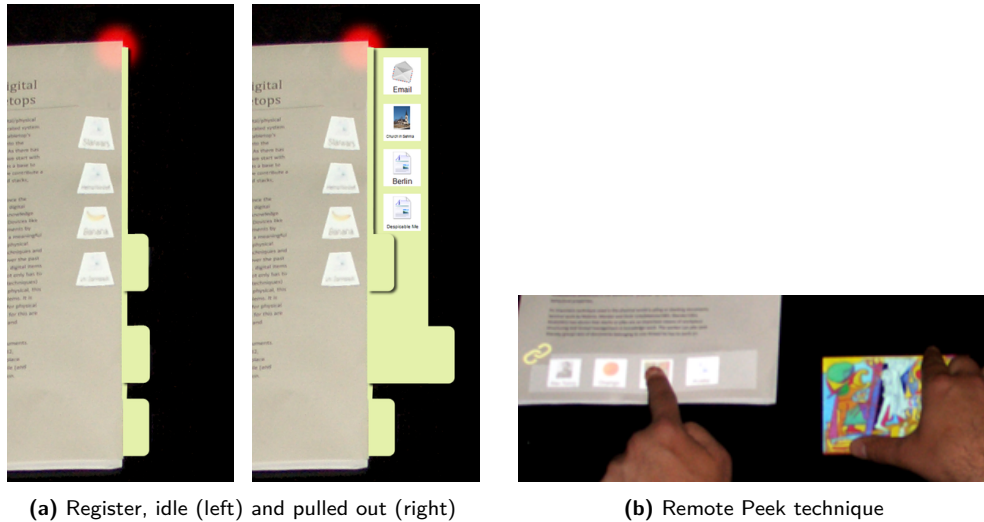


Figure 4.6: Loose coupling visualizations.

In-Place Access In-place access allows the user to directly access the digital documents stacked above the topmost physical document by showing them as a stack of icons in a free area in the margin of the document (see figure 4.5). The user can look at the documents by dragging them off the stack, resulting in the full object being shown on the tabletop. Further, StackTop provides a remote peeking gesture, a bimanual gesture in which a single-finger touch references a target object (e.g., the icon shown on the document), and the other hand can span a display area with two fingers for temporary inspection (see figure 4.6b).

Registers Registers use the register metaphor known from physical folders and enable local “at-the-stack” access to digital content even at positions within the stack that are not directly visible by pulling out the corresponding register. Each register represents a layer of digital documents between two physical documents ordered from top to bottom (see figure 4.6a). Using the registers, it is possible to peek at, add, and remove digital content at arbitrary levels of the stack.

StackPath StackPath gives a complete overview of the stack. The contents of the stack are lined up equidistantly along a *user-drawn coherent path* starting at the stack indicator. During drawing, the positioning of the documents along the path is updated continuously. When lifting the finger, the path with the documents lined up remains visible for further inspection by the user. The

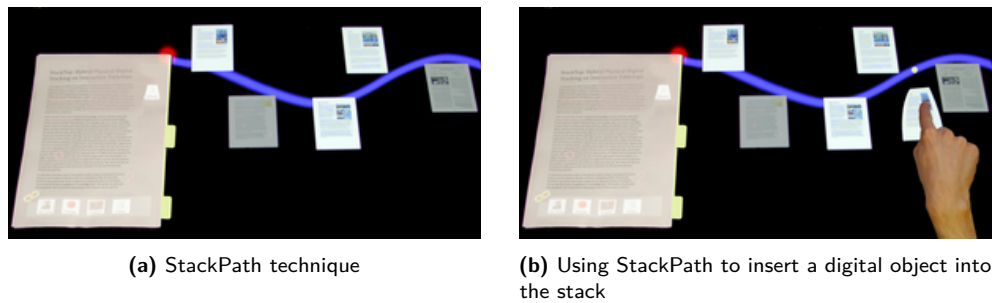


Figure 4.7: Path-based stack access.

user can thereby optimally use the free space on the tabletop without moving other objects.

Physical documents are shown slightly dimmed to signal that not all actions can be performed. For instance, it is not possible to remove a physical document from the stack using the StackPath visualization. Otherwise, StackPath supports the same interactions as the registers: Dragging out a document results in a copy (this is also possible for physical documents, using a stored digital image), dragging a digital document at any position inserts it at the selected position within the stack (see figure 4.7b), and a direct HyperMove gesture performed on a digital document actually removes the digital item the gesture is performed on (see figure 4.7a).

Further, StackTop provides a global full-text-based search function for digital and physical documents, highlighting their occurrences at the stack and document levels by showing an outline around the stacks containing relevant documents. Again, just as for the Content Access for hybrid documents, this requires a digital copy to be available.

Reordering

The last operation supported by StackTop is reordering. Depending on the scope of the reordering operation, the user has several options. To reorder documents within a specific digital layer, the registers can be used: By dragging the icons within the register, it is possible to alter their order. The same holds for the in-place access on the topmost document. To move digital documents within the stack across physical documents, one can remove a document from one register by using direct HyperMove to place it onto the tabletop and then add it to another register.

A more convenient and direct way is to use the StackPath: The StackPath

Name	Purpose	Description
Stack Building	Building a stack of documents	Adding documents to a stack by moving them onto an existing stack or single document.
Stack Building	Decomposing a stack of documents	Removing documents from a stack by HyperMoving them out of a stack.
In-Place Access	Accessing digital documents	The topmost digital documents are directly visible through an icon visualization in the free margin of the topmost physical document. It can be used to reorder digital documents via drag-and-drop or to inspect them by using the remote-peek gesture.
RemotePeek	Viewing details of digital documents	Temporarily peeking at a digital document by referencing an icon with a single finger and spanning an area with two fingers of the other hand in a free area.
Registers	Accessing intermediate layers	Pulling an intermediate digital layer out of the stack and providing access to the documents placed there via an icon representation.
StackPath	Displaying an overview of the stack	Providing an overview of the stack in which all documents within the stack are aligned along a user-drawn path.
Reordering	Moving documents within the stack	Using In-Place access, Registers, and StackPath to reorder documents within their respective scope.
Copy-On-Drag-Out	Accessing a document temporarily	Dragging a digital copy of any object within the stack out for deeper short-term inspection without affecting the stack.

Table 4.3: Summary of the interaction techniques for hybrid stacks included in the StackTop interface.

allows for intuitive reordering of the stack by dragging the documents around. However, as the system cannot move the physical documents, they are fixed in their position, serving as a skeleton.

In order to move a physical document, one has to directly interact with the stack as one would with a conventional physical stack as described above. When removing a physical document, the digital documents that were on top of the physical document remain in the stack. When the physical document is put back into the stack at another location, it is placed on top of the digital documents that are at that position.

Copy-On-Drag-Out

As already mentioned in the description of the techniques above, in order to facilitate temporary access to documents from a stack, StackTop supports a *Copy-On-Drag-Out* behavior. Upon dragging a document from a Register or StackPath overview, a copy of the document is created instead of removing the actual object. To actually remove it, the user can use the direct HyperMove gesture by referencing an empty spot on the tabletop and flicking the digital document out of the register or path. This allows temporary access to documents at intermediate levels without altering the stack.

4.4 User Study

<i>Method:</i>	Qualitative	In order to see how users perceive the StackTop concept for hybrid stacking, we conducted a first qualitative user study.
<i>Participants:</i>	7	
<i>Duration:</i>	ca. 40 min	
<i>Data Gathering:</i>	Questionnaire, semi-structured interview	

Table 4.4: Study Design Overview

Goal The goal of the study was to determine how users would use hybrid stacking when there is a system supporting this concept.

We were interested to see how well users could grasp the concept of hybrid stacking and how the techniques StackTop offers perform in supporting the users. More specifically, we were interested in the following two research questions:

1. Do users employ their natural practices of stacking in a true hybrid environment where they can intertwine both physical and digital documents?
2. How do users perceive the concept of hybrid stacking and how do they perform hybrid stacking in practice?

Participants For the study, we recruited 7 participants from our lab (all male, aged between 22 and 33, $\mu = 26.7$). All had at least occasional previous experience with tabletops. They were not given any compensation for participating in the study.

Apparatus and Setup To conduct the study, we implemented the described techniques in a fully functional C# application using DirectX for visual output. Again, the implementation was based on the PeriTop setup described in chapter 2 and took place in our lab. During the study, a set of physical objects was present on the tabletop (printed papers, Wikipedia articles, and a book) as well as some digital objects (digital Wikipedia articles, images, and notes). The users were asked to play around with the system. We did not impose a specific task besides experimenting with the system; therefore, the arrangement of the objects was up to the user.

Experimental Design As we were interested mainly in the general perception of hybrid stacking, we let the participants freely use and explore the system. The participants were given an introduction to the system and were allowed to play around with it for about half an hour. During this time, they were encouraged to tell anything they found (think-aloud style). Afterward, they were asked to answer a set of questionnaires: NASA TLX to assess the cognitive load of our hybrid stacking environment, as well as SUS and AttrakDiff to get additional feedback on the usability of the system. For the TLX, we omitted the temporal demand question because we did not impose any time restrictions and because of some stability issues with the touch detection on objects.

4.5 Results and Discussion

This section presents and discusses the results of the user study.

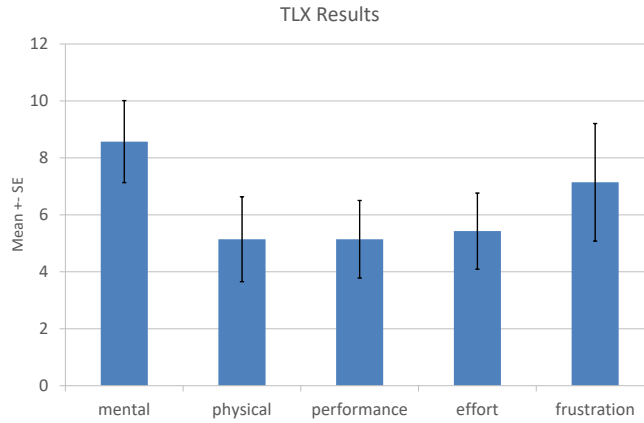


Figure 4.8: TLX results. Full scale ranges from 0 to 20.

4.5.1 TLX

The NASA TLX revealed that using StackTop imposes a sub-average load on the user with all values being below 10 (see figure 4.8). The relatively high value for mental demand can be explained as a result of the novel concept, which the users had to grasp first. However, as it is still below 10, one can deduce that the concept can be easily understood by casually using the system. The value for frustration can be explained mainly by the touch issues, which is backed by the feedback users gave during the sessions. These issues were caused by interference between the depth camera and the PixelSense technology, and we therefore consider it an implementation issue.

4.5.2 SUS

The SUS questionnaire revealed a score of 74.28 ($\sigma = 8.93$) for our system, which is above average and indicates good overall usability.

4.5.3 AttrakDiff

The AttrakDiff questionnaire indicated a very high hedonic quality ($\mu = 1.5$, $\sigma = 0.38$) as well as an above-average pragmatic quality ($\mu = 0.89$, $\sigma = 0.95$). The comparatively low value for pragmatic quality is a result of the system being perceived as more technical ($\mu = -0.42$, $\sigma = 1.49$) as depicted in figure 4.9a. One reason for this might have been our lab setup with its unnatural environment and the aforementioned touch issues. The attractiveness score also showed a high value ($\mu = 1.67$, $\sigma = 0.79$), indicating that users like the

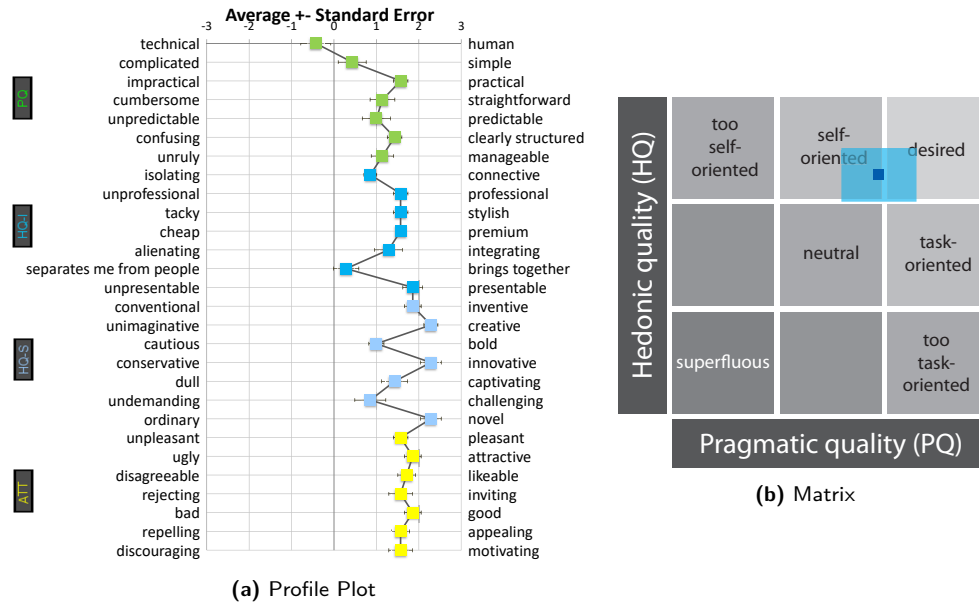


Figure 4.9: Results from the AttrakDiff Questionnaire for ProjecTop.

system. Overall, the system can be described as creative, innovative, and novel based on the highest ranked attributes (cf. figure 4.9a).

4.5.4 User Feedback

All participants liked the idea of hybrid stacking very much. They enjoyed stacking digital with physical objects, e.g., by creating alphabetically sorted stacks that would be impossible to build without stacking support.

Despite the overall positive results, participants also had some critical feedback regarding some of the features StackTop provides.

Projection on Empty Places

Using empty areas on the document in order to use physical documents as handles or carriers of digital information was intuitive to all participants, with two participants explicitly mentioning it as positive (*“Using physical objects as a carrier for digital content is very nice”*, P4). However, P1 noted that using free areas might be problematic if the user explicitly wants the area to be free and uncovered. This is a general issue with systems projecting on objects not explicitly made for projection, as the projection can always interfere with the user’s intent. A possible, yet somewhat inconvenient, solution is to allow

the user to mask specific areas, which then are not used to project content. Unlike the situation in the context of ProjecTop 3.7, the feedback indicated no difference between working and storage areas concerning the acceptability of projecting onto free areas of the document. P1’s concern is of a general nature and applies to all physical documents regardless of their position.

User-Drawn Path for Exploration

The *StackPath technique* was well received; however, two of the seven participants found the current way of closing the path (tapping on the red dot on the stack again) too time-consuming. The reason for this is that while we initially thought of StackPath as a means of reorganizing and deeply viewing the stack, participants also used it to get a short glimpse at the contents of a stack (e.g., P2 said “*I just want to quickly look which elements are in the stack, and not interact with the StackPath in any way.*”). In order to better support this use case, participants suggested allowing users to close the path with a single tap on a free area on the tabletop. However, this solution (and other solutions that rely on tapping) might lead to problems in the case of unintentional tapping (e.g., playing around with the idle hand on the tabletop while looking at the path or reading a document). A possible solution could be to use a more complex, yet fast, gesture such as “drawing” a circle.

Copy-On-Drag-Out Behavior

The default *copy-on-drag-out* behavior for registers and StackPath was received very negatively. Five of the seven participants preferred a conventional behavior as the default, i.e., that the element is actually removed instead of creating a copy (P1: “*Because the system is on a tabletop, one expects the digital documents to be used just like physical documents - the copy-on-drag-out technique doesn’t fit into this concept*”, and “*It is more important to alter a stack than not to do it.*”). Copy-on-drag-out was, however, noted to be very useful in some cases, but should then be accessed through a special gesture (e.g., a two-finger pull-out gesture).

Registers

The *register metaphor* to access specific layers of a stack was found to be very practical by most participants. Two participants criticized the visualization

style as suboptimal. They found the registers insufficient to depict the structure of the stack because the exact location (absolute position) of a register within the stack and the number of digital objects at the given position are not visible. They suggested adding a more three-dimensional representation (e.g., displaying virtual pages between the registers to denote their thickness) to overcome these issues. One participant even wanted direct access to all objects in the stack without having to open a register or use the StackPath.

Extensions

Participants also made suggestions about how to extend the system: P5 wanted to add a printer to the tabletop so that digital objects can be printed ad hoc and then used as physical objects. P4 wanted a semantic-based visibility control that hides all digital content not relevant to the physical document currently in the focus area.

Summary

In summary, the StackTop concept of hybrid stacking was generally very well received and users enjoyed using it. Compared to the approach used by ProjecTop in the previous chapter, the use of free spots on the documents was appreciated. Although a few of the design decisions were criticised, especially the default use of copy-on-drag-out, they are not fundamental to the concept and can be easily resolved in the next iteration. Further, the participants made valuable suggestions for improvements or extensions for the next iteration of the system.

4.5.5 Conclusion

In this section, we presented the results of a first study on hybrid stacking for interactive tabletops, which have shown a high potential for hybrid stacking. Based on the results, we can answer our initial research questions as follows:

Do users employ their natural stacking practices in a true hybrid environment? Based on the study, the answer is definitely yes. The AttrakDiff results and the users' reactions clearly show that hybrid stacking is well received and users like to use it.

How do users perceive the concept of hybrid stacking and how do they perform? We initially thought users might have problems with the fact that digital content can now be between two physical objects and not only on a specific digital surface. However, the study has shown that users do not have any issues with this and that they casually use hybrid stacks without much (mental and general) effort.

The feedback indicated some design issues with the current techniques offered by StackTop that can be easily solved (e.g., changing the behavior to a more physical-like one).

4.6 Implications for Hybrid Stacking

Based on the results of our study, we derived a set of design implications for systems supporting hybrid stacks.

4.6.1 Digitally or Physically Oriented Interaction?

While we initially thought that relying on the possibilities of digital documents to augment hybrid stacking (e.g., providing the *copy-on-drag-out* behavior) would be beneficial for both digital and physical documents, our study clearly showed that users expect and want the hybrid stack to behave more like a physical stack. Nevertheless, the digitally inspired techniques were rated as useful in special situations. As a result, techniques providing non-physical behavior of objects within the stack should be made accessible through special gestures. The standard touch interaction gestures (e.g., one-finger drag) should lead to behavior known from the physical world.

4.6.2 Visualizing the Stack

Visualizing a hybrid stack is a challenging task because the physical part has a physical height, while the digital part does not. Ideally, one would be able to visualize additional sheets of paper at the location the digital documents are placed within the stack. Currently, this is not possible, but one can choose visualization techniques that give some height clues to the user. For example, our register-metaphor could be extended by displaying documents between the registers. The registers would then be irregularly spaced according to their thickness, giving more information about their position compared to the

current even spacing independent of the number of documents.

4.6.3 Accessing the Stack

Accessing or browsing a stack is one of the more tedious tasks. Hence, digital augmentation is well received there. Depending on the use-case, access to either the whole stack (as in our StackPath technique) or a subset (like our registers) is desired by users. Systems should, therefore, be able to accommodate both styles of access. Based on the general user feedback, it may even be desirable to expand the concept to context-sensitive techniques, like automatically highlighting stacked documents based on the document the user is currently working on (e.g., the document within the user’s working area).

4.7 Summary and Conclusion

In this chapter, we presented the StackTop concept, an integrated approach to support hybrid stacking on digital tabletops. It bridges the gap between physical and digital objects, beyond simply resolving occlusion in a general manner, by explicitly modeling stacks. Thereby, the lack of support for stacking in hybrid environments described in the motivation chapter of the thesis is addressed. This can be seen as another important step towards closing the gap between digital and physical documents in the context of hybrid work environments: One of the most important organization strategies is thereby expanded from the physical domain to use in hybrid scenarios. With respect to the ProjecTop concept for resolving occlusion, StackTop can be seen as a complimentary approach that can be used alongside the ProjecTop concept. In this case, one can imagine that StackTop would be used for document objects (cf. section 2.3.2) being placed on top of each other while other objects being placed on top of documents would lead to a ProjecTop-style visualization.

The StackTop concept allows users to easily build and work with hybrid stacks by providing means of awareness and access to the documents within the stack. Additionally, it supports not only loosely coupled stacks of documents (“*hybrid stacks*”) but also tightly coupled content at the document level, like annotations or linked documents (“*hybrid documents*”), which is comparable to adding a physical Post-it to a document or stapling documents together in the physical world. In order to implement the StackTop concept, the PeriTop approach of peripheral display is leveraged again to visualize digital content in

place to improve digital/physical integration.

A first user study conducted using the prototype has shown that the concept of hybrid stacks is easily understood and well received by users. Contrary to the unconditional projection onto objects employed in the ProjectTop concept, the use of free spots on the documents has been generally appreciated by users. Thus, it can be seen as a promising way of employing top-projection in the context of hybrid work environments. The next chapter introduces an approach for automatically finding such free spots.

Part III

Layout for Hybrid Environments

FreeTop – Solving the Placement Problem

The StackTop concept introduced in the previous chapter uses free spots on documents to display its visualizations. A question that remained unanswered is how these free spots can be identified automatically. This chapter proposes FreeTop, an approach to deriving a per-pixel projectability measure based on multiple factors that can be used to identify suitable free spots on documents. Beyond the application to documents in the context of hybrid workspaces, other domains, such as augmented reality, also face the problem of interference between the texture of physical surfaces and digital projection. Thus, they can also benefit from the FreeTop concept.

The chapter is structured as follows: After an introduction (section 5.1), a review of relevant related work is presented in section 5.2. Section 5.3 then introduces the projectability measure and explains its calculation and properties. The chapter concludes with an overview of the technical background of the implementation of FreeTop (section 5.4) and a summary of the chapter (section 5.5).

5.1 Introduction

In the previous two chapters, we argued that the surface of occluders could be leveraged as a display area. In order to do so, suitable areas on the surface of an occluder are needed that have a sufficient size and ideally as little texture as possible. Thereby, the risk of interference between the texture of the occluders and the projection is reduced. This is especially important in the context of StackTop, which continuously displays complex digital content, like the stack- or link-bar, on physical objects. Therefore, the question of how to find areas suitable for projection remains.

Orthogonal to the interaction techniques and concepts described in the

context of this thesis, the problem of finding appropriate areas for projecting content also occurs in many other contexts. For instance, with the advent of inexpensive projection technology, interactive projective augmentation of the physical world has become a common technique in the field of HCI [Hartmann et al., 2010; Liao et al., 2010; Wellner, 1993]. Typically, projection is used to digitally augment surfaces and everyday objects, for instance, by displaying additional information [Liao et al., 2010] or self-contained digital content [Cotting and Gross, 2006; Huber et al., 2012]. While practical and widely used to bridge the digital and physical worlds, one prominent issue of projection on objects is interference between the texture (physical and visual) of the physical surface and the projected digital content. Therefore, it is desirable to automatically find areas in the physical world with properties that facilitate projection and interaction (e.g., untextured, light, flat surfaces).

One way to mitigate this issue is to use predefined models of objects that denote spots suitable for projection, in the simplest case by manually specifying the coordinates of the projection areas. This obviously requires all objects to be static and known in advance. For example, in the context of StackTop, the prototypical implementation assumed the margins of the documents to be empty, and the documents were chosen such that this assumption held true. However, in practical use, such assumptions may not always hold, and unknown objects might be used for which no projection areas are defined. Thus, more sophisticated approaches are required.

A more dynamic practically used approach is to use corner-detection [Liao et al., 2010] to find areas covered by content (e.g., text). Using corner detection to assess projection quality works well with text, but other content, e.g., pictures, might not have corners to detect. One solution is to use structured light [Cotting and Gross, 2006] to find areas suitable for projection. This circumvents the limitations of corner detection but requires significant technical effort. Both approaches suffer from the fact that they do not take the three-dimensional surfaces of objects into account. This can lead to issues with the legibility and interactability of the projection: For instance, a white box on a white table may appear as smooth surface from a top-mounted camera, but projection over the edge makes it quite difficult to recognize the content and interact with it. Also, extremely rough surfaces might be unsuitable for touch interaction, but look smooth on the camera image.

To overcome the limitations of related approaches, the FreeTop approach uses a commodity RGBD camera (e.g., Kinect) instead of a normal RGB

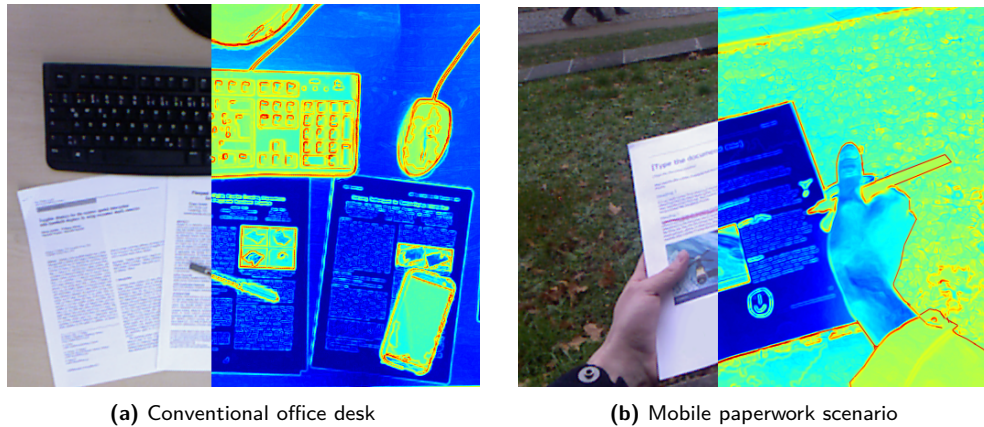


Figure 5.1: Colorized projectability map (dark blue means good projectability, red means bad projectability) of two example scenarios.

camera. This makes it possible to combine color and depth information, which allows gaining additional insight into the surfaces of objects and a better understanding of the overall projection surface. This results in two major advantages:

1. By combining color- and depth-based (surface lightness, visual and physical edges) measurements, a *projectability score* can be assigned to every pixel within the camera image, leading to a projectability map of the scene.
2. The generated projectability map of a given scene or object can then be used to inform the layout and placement of projection.

The FreeTop approach can be employed in stationary or mobile (e.g., nomadic pico projectors [Huber et al., 2012; Kane et al., 2009] or AR glasses) settings for on-surface augmentation. We primarily focus on stationary office settings without limiting FreeTop to this area (see figure 5.1 for an example of mobile use).

5.2 Related Work

Generally, the requirement of empty spaces for display purposes is common in a wide range of applications. In the following, we discuss currently used approaches in the most prominent applications. Although the works on augmented desktops and occlusion management have already been discussed in

previous chapters, they will be discussed again in the following with a special focus on free spot finding.

5.2.1 Augmented Desktops

As outlined in section 2.2.1, many efforts have been made to create *augmented paperwork* with digital projection or input, thereby allowing the user to add digital content interactively. As a result, researchers have also, to some degree, integrated means to detect locations for the placement of their digital augmentation.

However, most of these systems aim at augmenting paper documents with additional digital facilities and do not focus on self-contained digital objects (e.g., a fully digital document). To some extent, these systems take possible interference between physical and digital content into account (e.g., FACT [Liao et al., 2010] uses corner detection to identify areas covered with text). However, most of the digital content targeted in these works is related to the physical object it is projected on. For instance, the Paper-top interface [Mitsuhara et al., 2010] augments animations onto learning material for illustrative purposes. As a result, a certain amount of interference is tolerable since the annotations, for example, are deliberately placed by the user. To reduce interference, manual user-defined projection areas have been considered [Xiao et al., 2013]. This, however, requires active user intervention and does not automatically account for changes in the surface.

For general-purpose tabletop systems, approaches to finding an uncluttered area suitable for displaying content were proposed [Siriborvornratanakul and Sugimoto, 2008]. The Display Bubbles [Cotting and Gross, 2006] system uses bubble-shaped display areas to fill the areas between physical obstacles and thereby avoid interference. Both of these approaches aim at finding large areas for projection and inherently avoid projecting on objects, as the presence of objects is used to mask the display area. As a result, they are not suitable for augmenting, for example, individual documents.

In scenarios where the digital content should be dominant, color correction [Fujii et al., 2005; Grossberg et al., 2004; Nayar et al., 2003] can be used to allow uniform projection on textured surfaces. The goal of these approaches is to adjust the projected image so that it compensates the different colors of the projection surface to yield a uniform visual appearance despite a non-uniform projection surface. However, while color correction mitigates the problem of

interference in favor of the digital content, it has some limitations when it is not possible to compensate the texture of a surface – for instance, printed black text on a white background cannot be made invisible. As a result, strategies are still required to avoid projection on certain areas.

Research investigates content awareness by avoiding projecting onto textual content either automatically [Cotting and Gross, 2006; Liao et al., 2010; Siriborvornratanakul and Sugimoto, 2008] or through user intervention [Xiao et al., 2013]. To avoid projection on content, it is common to either analyze the surface texture (e.g., with an RGB camera) to avoid textural interference [Liao et al., 2010] or to analyze the physical structure of objects [Cotting and Gross, 2006] to avoid physical interference. Based on these results, it is common to either relocate content to a free area [Liao et al., 2010] or transform the representation to fit the free space [Cotting and Gross, 2006], or combinations thereof.

These approaches all rely on a binary projection quality assessment, masking out areas deemed unsuitable and thereby losing the information about how unsuitable the areas actually are.

5.2.2 Occlusion Management

In a similar direction, *occlusion* has been identified as a problem when working with physical objects in hybrid tabletop scenarios [Steimle et al., 2010b]. As a result, user-driven layouts have been explored to circumvent obstacles and occlusion for menus [Leithinger and Haller, 2007]. Also, systems have been developed to automatically avoid occlusion on tabletops for labels [Sato and Fujinami, 2014] or general objects instead of menus [Furumi et al., 2012; Khalilbeigi et al., 2013]. Again, these systems rely on a binary occlusion measure and do not trade off between closeness to the original location and display quality. This can lead to situations in which an object is displayed far from its intended location as no closer large-enough free spot is available.

All approaches require finding areas suitable for display, whether it is finding a free spot *anywhere* on the surface to relocate the item, *next to* a physical object to display the proxy, or *on* the physical object. Most systems tend to simply assume that the area right beside the occluder or the margins of an object – as they are commonly blank – is a suitable place. However, this assumption does not always hold: The surface might be dark, very textured, etc., rendering any content displayed unreadable. The messy tabletops concept

[Freeman and Brewster, 2013] allows for finding a large free area on a back-projected tabletop. However, this approach only considers back-projected tabletops, and neither provides information for top-projection setups nor takes into account the surfaces of objects.

5.2.3 Tangible User Interfaces

For *tangible user interfaces*, projective augmentation is of particular importance. One common approach is to augment three-dimensional objects by tracking them and projecting onto them using sensors in the environment. Thereby, visual content can be augmented onto a physical environment [Jones et al., 2010; Wilson, 2007, 2005], onto moving objects [Khalilbeigi et al., 2012; Steimle et al., 2013], or next to tangibles [Funk et al., 2014]. Further work explored the use of the surroundings of an object to change its look [Lindlbauer et al., 2017] or provide additional context [Underkoffler and Ishii, 1999]. It is noteworthy that in the context of these related works, it is often simply assumed that the surface of objects is uniform and suitable for projection [Funk et al., 2014; Khalilbeigi et al., 2012; Steimle et al., 2013] and that projection happens directly on or around the tangible in a static manner [Funk et al., 2014; Underkoffler and Ishii, 1999].

5.2.4 Augmented Reality

In the field of *augmented reality*, the placement and layout of visual overlays is an important issue as well. Grasset et al. [2012] propose an image-based approach using a canny edge detector combined with a thresholded saliency map (leading to a there-level importance map) to classify the visible area, looking for areas with no edges and a low saliency. This approach, however, limits insight into the readability of the overlay by using binary edge detection and focusing only on the importance of the content. However, a bright white (or dark black, which is unsuitable for projection) surface has a low saliency and no edges, but the overlay won't be easy to read, which cannot be detected by this approach.

In a similar direction, Orlosky et al. [2013] developed another approach to locating dark, billboard-like structures in an image on which to overlay text. This method employs the pixel brightness and its standard deviation over the size of the object to be displayed to ensure uniformity. The Hough lines algorithm is used to ensure stability; however, it is not used for the projectability

calculation. Using standard deviation as a feature is somewhat problematic as it does not take into account the structure. A rectangle containing half black and half white pixels will always have the same standard deviation whether the upper half is black and the lower white or the pixels are randomly distributed. However, there is certainly a difference when it comes to readability between these two.

Based on predefined or generated 3D models, research has emerged to inform the layout of AR environments. For instance, free and occupied spaces can be modeled by using rectangular areas, which are then used to lay out content [Bell et al., 2001; Bell and Feiner, 2000].

Using RGBD cameras, Ens et al. [2015] proposed an approach to spatial constancy, using a Kinect-fusion-based [Izadi et al., 2011] model together with a saliency map.

However, both approaches require generating a complete 3D model of the whole scene to identify the locations of the objects in the target workspace area. Gal et al. proposed a constraint-based optimization approach for AR layout generation [Gal et al., 2014] that considers the position and scaling. However, they do not consider interferences with physical objects/surfaces and therefore do not take into account the quality of the projection or overlay. While this is sufficient in an ideal AR world with opaque overlays, current implementations of AR glasses or projection-based AR systems are semi-transparent, and hence interference should be accounted for. Setting aside the technological possibilities of fully opaque overlays, there are scenarios in which transparency is even desired. For instance, when using AR to guide a worker during a machine repair, it could be fatal if opaque overlays covered dangerous machine parts or limited the worker’s free field of sight.

5.2.5 Requirements and Summary

In the following, we present a set of requirements for free spot finding derived from reviewing the related work and use them to compare the most relevant related work (see table 3.1).

- R1 **Texture interference detection** The visual texture of an object is one of the main factors affecting projection quality (e.g., very dark or strongly structured surfaces are unsuitable for projection). Hence, systems must take it into account.

	Texture Interference Detection	Physical Interference Detection	Continuous Measurement	Relocate Content	Transform Content
<i>Augmented Paperwork and Desks</i>					
Cotting and Gross [2006]	●	●	○	○	●
Liao et al. [2010]	●	○	○	●	○
Mitsuhara et al. [2010]	○	○	○	○	○
Xiao et al. [2013]	○	○	○	○	○
<i>Occlusion</i>					
Furumi et al. [2012]	○	●	○	●	○
Sato and Fujinami [2014]	○	●	○	●	○
Khalilbeigi et al. [2013]	○	●	○	●	●
Freeman and Brewster [2013]	○	●	●	●	○
<i>Tangibles</i>					
Funk et al. [2014]	○	○	○	○	○
Underkoffler and Ishii [1999]	○	○	○	○	○
<i>Augmented Reality</i>					
Bell and Feiner [2000]	●	○	●	○	
Ens et al. [2015]	●	○	●	●	○
Gal et al. [2014]	○	○	○	●	●
Grasset et al. [2012]	●	○	●	●	○

Table 5.1: Properties of a representative selection of layout systems. ● indicates that a requirement is fulfilled, ● indicates partial fulfillment, and ○ that a requirement is not fulfilled.

- R2 Physical interference detection** Similarly, the physical structure affects projection quality (e.g., rough, uneven surfaces yield inferior quality to flat surfaces).
- R3 Continuous measurement** The impact of interference is not binary, but a continuous scale. Hence, approaches should model their assessment on a continuous scale as well.
- R4 Relocate content** In order to improve projection quality, systems should provide a way to find suitable locations for projection in order to relocate content to more suitable areas.
- R5 Transform content** In order to further increase quality, not only relocation should be supported but other transformations, like scaling or rotating, should be facilitated as well.

In summary, while suitable for their respective target environments, the existing approaches have drawbacks when projecting on physical objects. To avoid interference, they often search for uniform surfaces in terms of color (e.g., low saliency, no edges or corners, etc.), neglecting their physical structure. For projection, physical edges, optically invisible in the camera image, can cause problems. Also, dark surfaces are often not well suited for projection despite being uniform. Moreover, when adjusting the presentation of the content to adapt to the environment, existing approaches often either relocate or transform content, but not both. In contrast to prior work (see table 5.1 for a comparison of the most relevant works), FreeTop expands on this and takes into account physical features like surface smoothness or edges as well as surface lightness to assess projection quality by employing a continuous projectability measure.

5.3 FreeTop Projectability Map

To overcome the limitations of previous approaches and to enable a more flexible and appropriate placement of projected content, we present a method to assess the visible surface in a scene regarding its suitability for projection. Instead of doing this in a binary way (projectable/not projectable), we assign a *projectability score* to each pixel. For practical reasons, the score ranges from 0 (very suitable) to 255 (unsuitable). This range was chosen so that the maps can be internally processed as an 8-bit greyscale image. Therefore, it is

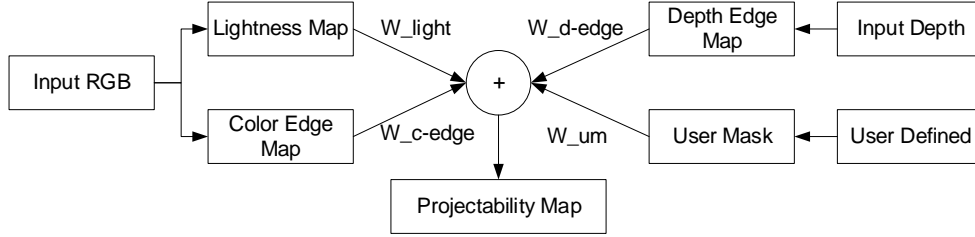


Figure 5.2: Abstract structure of the projectability map calculation.

directly processable by standard image processing libraries. Areas usable for projection are, in general, of *light color* (or dark for AR-Overlays) and ideally *smooth*, without strong visual texture or content (e.g., text). We therefore consider the lightness and smoothness of the surface, similar to Thanedar and Höllerer [2004] and Tanaka et al. [2007]. If the display should be interactive, a *physically continuous* and smooth surface is also desirable. FreeTop takes these factors into account and allows flexible weighting of the different aspects. This allows for more refined and flexible placement rules (e.g., it might be acceptable to have a worse projection quality if the projection is closer to the originally intended location). As depth cameras usually provide their intrinsic calibration data, only calibration of the output device (e.g., projector) relative to the camera is needed and must be maintained by the high-level application. To account for variability in brightness/contrast or ambient light, FreeTop employs automatic white balance and brightness correction.

5.3.1 Projectability Map

As depicted in figure 5.2, FreeTop generates four maps, described in the following, which are then weighted and added to form the final projectability map. The actual weights are derived from user-specified weights ranging from 0 to 100 for each map by normalizing their sum to 1. This leads to a final projectability map. All maps are derived directly from the input camera image and thus have the same resolution as the input depth and visual image. Since they are overlaid during the process of building the final map, both depth and color image need to have the same resolution. In our case, the depth input images is 512×424 pixels in size, hence the color image is scaled accordingly. Internally, the individual and final maps are all processed as 8-bit grayscale images.

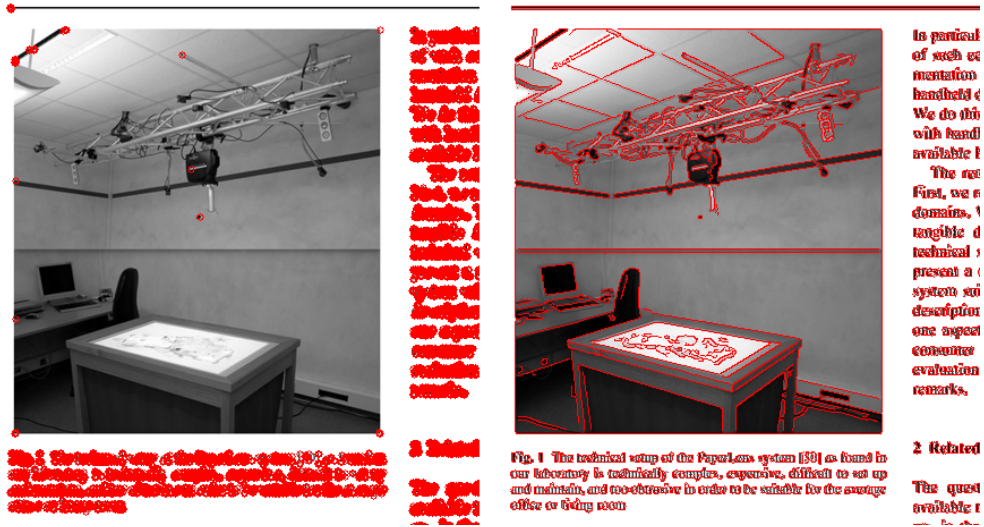


Figure 5.3: Comparison of corner detection (left) vs. edge map (right): corner detection does not respond well on an image.

Smoothness

To locate any obstacles such as text, figures, etc., we choose an edge detector instead of the common corner detection approach [Liao et al., 2010]. While corners are suitable features for printed text, other content can be problematic (see figure 5.3 left): shapes without prominent corners, like lines, circles, etc. cannot be captured by a corner detection approach. As a result, these areas would be considered by the algorithm as suitable for displaying content when they are actually not.

In contrast, an edge detector not only captures all shapes that corner detection responds to but is additionally able to capture lines, circles and other shapes without corners (see figure 5.3 for a comparison). The output of edge detection is used as a *color edge map*. Conventionally, edge detectors, like Canny (e.g., as used by Grasset et al. [2012]), are binary (i.e., they return 1 if there is an edge and 0 if there is not).

However, as we target a continuous assessment of projectability (e.g., an edge between a yellow and a white surface is not as bad as a hard black/white contrast), we use a gradient-magnitude-based approach, which allows for a continuous response for edges. The gradient is derived using the Sobel operator in the x and y-directions. The magnitude is then the length of the resulting vector. Listing 1 shows the pseudo-code for the calculation.

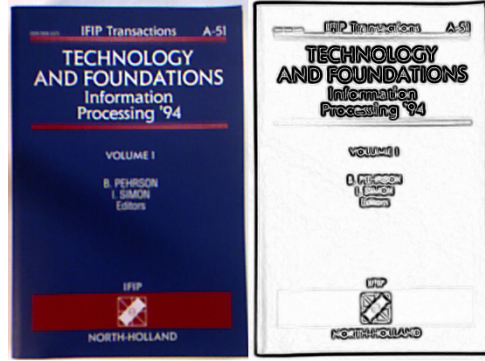


Figure 5.4: Example of dark areas, not recognized by edge detection.

Algorithm 1 Calculation of the color edge map.

```

sobelx  $\leftarrow$  SobelX(RGBImage)
sobely  $\leftarrow$  SobelY(RGBImage)
for  $x \leftarrow 0, x < \text{Width}(\text{RGBImage}), x \leftarrow x + 1$  do
    for  $y \leftarrow 0, y < \text{Height}(\text{RGBImage}), y \leftarrow y + 1$  do
         $\text{cemap}_{x,y} \leftarrow \sqrt{\text{sobelx}_{x,y}^2 + \text{sobely}_{x,y}^2}$ 
    end for
end for
return cemap

```

Lightness

Even though the edge-based approach is able to capture a wide range of possible obstacles, there is still a problem with (very) dark (or white for AR), yet smooth, surfaces that do not contain any edges (see figure 5.4). As mentioned, projection is more suitable for lighter surfaces. We therefore consider the surface lightness in order to find suitable spaces in the form of the *lightness map*. It is defined as the inverse of the normalized and thresholded grayscale version of the color image. Listing 2 shows the pseudo-code for the calculation.

Algorithm 2 Calculation of the lightness map.

```

gray  $\leftarrow$  Grayscale(RGBImage)
grayN  $\leftarrow$  Normalize(gray)
lmap  $\leftarrow$  Invert(grayN)
return lmap

```

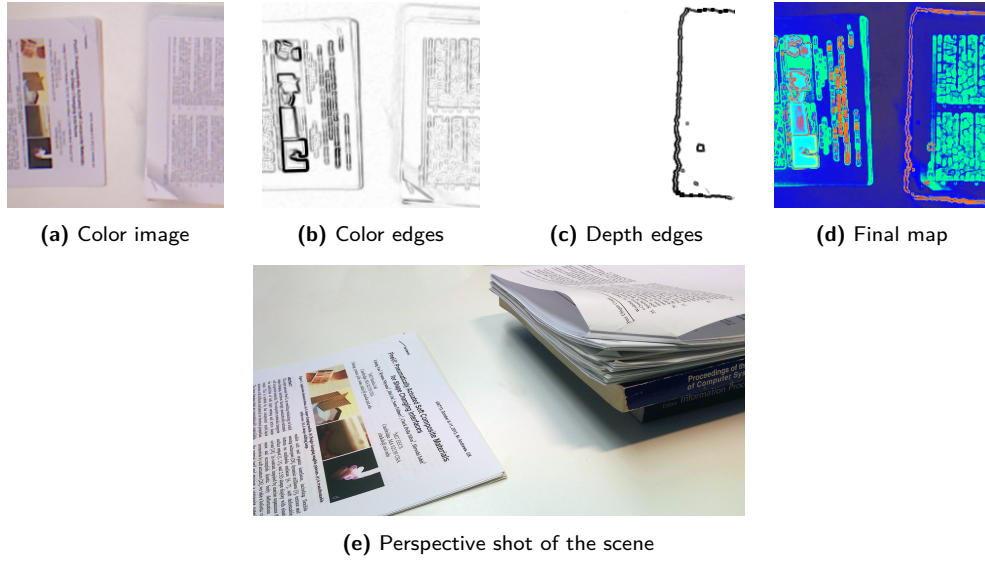


Figure 5.5: Physical edges are not always recognized on the color image.

Physical Surface

In addition to surface smoothness and lightness based on the color image, we also factor in the physical surface. This is important because the user has a different viewpoint than the camera. Hence, the surface might seem flat and continuous (i.e., no visible edges) from the perspective of the camera, but from the user's perspective, the surface is in fact not continuous (see figure 5.5). Therefore, FreeTop also considers the depth image to compute a *depth edge map* containing the physical edges of objects. Similar to the *lightness map*, a gradient-magnitude-based approach is used in order to factor in the height of edges. This is also important for interactive projections, as dragging, for instance, across a physical edge using touch is obviously impractical (c.f. chapter 3). Listing 3 shows the pseudo-code for the calculation.

Algorithm 3 Calculation of the depth edge map.

```

sobelx  $\leftarrow$  SobelX(DImage)
sobely  $\leftarrow$  SobelY(DImage)
for  $x \leftarrow 0, x < \text{Width}(DImage), x \leftarrow x + 1$  do
  for  $y \leftarrow 0, y < \text{Height}(DImage), y \leftarrow y + 1$  do
     $dmap_{x,y} \leftarrow \sqrt{sobelx_{x,y}^2 + sobely_{x,y}^2}$ 
  end for
end for
return  $dmap$ 

```

User Mask



Figure 5.6: The user mask avoids projection in the mouse area by adding a penalty.

Besides projection quality aspects, an application or the user might have additional personalized constraints (e.g., masking a certain area to avoid projection). These constraints can be considered in the form of a *user mask*, allowing the user to specify bias values for certain areas. For example, if no projection should occur within the user’s working area, the mask would assign the area a high penalty value (see figure 5.6 for an example). It can be specified either by directly drawing on the camera image or

by more advanced techniques implemented by the application using FreeTop (e.g., allowing the use of an actual paintbrush in the physical world).

Besides letting the user manually define the mask via drawing, an automatic generation based on customizable settings is also possible. For instance, as discussed in the context of ProjecTop (section 3.7), it could be used to model the working and storage areas inferred from the user’s behavior. The system would then assign penalties to the areas identified as working areas, thereby avoiding projection if possible. Since the user map is technically a grayscale image that is taken as input for the final computation, no pseudo-code is provided here, as it is application specific.

Final calculation

As outlined at the beginning of this section, the final map is calculated by weighting and adding the individual maps. Listing 4 shows the pseudo-code for the final calculation.

Algorithm 4 Calculation of the final projectability map.

```

 $w_{sum} \leftarrow w_{c-edge} + w_{c-light} + w_{d-edge} + w_{um}$ 
 $result \leftarrow cemap * w_{c-edge} / w_{sum}$ 
 $result \leftarrow result + lmap * w_{c-light} / w_{sum}$ 
 $result \leftarrow result + dmap * w_{d-edge} / w_{sum}$ 
 $result \leftarrow result + umap * w_{um} / w_{sum}$ 
return  $result$ 
    
```

5.3.2 Operating Modes

To support different scenarios, FreeTop can operate in two different modes, depending on the requirements:

Offline mode

The simplest mode of operation that can be used with FreeTop is the offline mode. In this mode, projectability maps are computed up front and remain static. Thereby, issues caused by interference between map computation and occurring projection are avoided since the maps are computed before the projection starts. Further, in this mode, the computation can also be done on data sources other than a live video feed from a camera. For instance, offline mode allows use of high-resolution images generated from the digital counterparts of physical objects for the visual maps and, for instance, digital 3D models for the depth edge map. Based on an external object tracking approach, the precomputed maps can then be used to inform projection onto the tracked objects. Also, the offline mode allows for better runtime performance as there are no runtime computations. Finally, the user can modify the projectability map for individual objects, e.g., through the UI of an application as the maps are available “offline”. A major drawback is that this mode does not account for texture changes occurring during application use, e.g., when drawing onto a document.

For example, in the context of StackTop (see chapter 4), offline mode could be used: StackTop already captures a cut-out image of the topmost physical document when it is placed. StackTop uses the image to identify the document and potentially match it with a corresponding digital copy. In offline mode, FreeTop could then be run once on the digital copy (providing a high-resolution projectability map of the document) or, if there is no digital copy, on the image captured by the camera. The resulting map can then be cached and reused whenever the document is visible and recognized.

Online mode

In this mode, a live camera image is analyzed either continuously or periodically to reflect changes in the environment. This mode is suitable in scenarios when very few or no specific objects are tracked individually and a complete overview of the whole scene is needed. Another use case is highly dynamic scenes in

which the layout and physical texture change frequently (e.g., many users are physically drawing on a large digitally augmented map). Also, in mobile ad hoc scenarios, this method is favorable, since the present objects are not known up front and the user usually moves over time rather than staying at a fixed position. As the projectability map is updated in real time, moving objects or changing surface textures of physical objects do not cause any issues.

However, since this mode processes live video data, interference with the projected content becomes an issue because it is seen by the camera and interpreted as visual texture on the physical object. As a result, the area that is projected on would be deemed unsuitable in the next iteration. Consequently, the projection would be moved to another location so that in the subsequent iteration, the original area would again be deemed suitable, causing the projection to again be located there and starting the process over again. In order to circumvent this problem, applications using FreeTop need to consider two possible scenarios: 1) The content they intend to project has already been projected at a place. Therefore, it is known that it was projected there before and the low projectability score is likely due to the current projection. Hence, it is safe to continue projecting there, as projectability was assessed before initially projecting. 2) The content is new and has not yet been projected. In this case, the interference is useful, as it invalidates the area currently used for projection for further projection (projections should not overlap).

5.3.3 Layout with FreeTop

Basic layout based on FreeTop can be done in several ways: The most trivial method is to search for the largest rectangle with a score below a threshold and then display all content within this rectangle (see figure 5.7a). However, more flexibility can be achieved by taking all areas with a score below a (dynamic) threshold into account (see figure 5.7b; note that projection across the object borders is avoided). If there are any soft preferences regarding the placement, the user mask can be used to accommodate them by providing a higher penalty for locations in which projection is unwanted (see figure 5.6; here the user prefers to have no projection around the mouse). Thereby, the masked areas are not fully ignored, but only used if there is no more suitable area. Using a mask instead of incorporating support for unwanted areas into the layout algorithm is beneficial because it reduces the complexity of the layout step and allows the user to, for instance, intuitively “paint” the areas where projection should not happen.

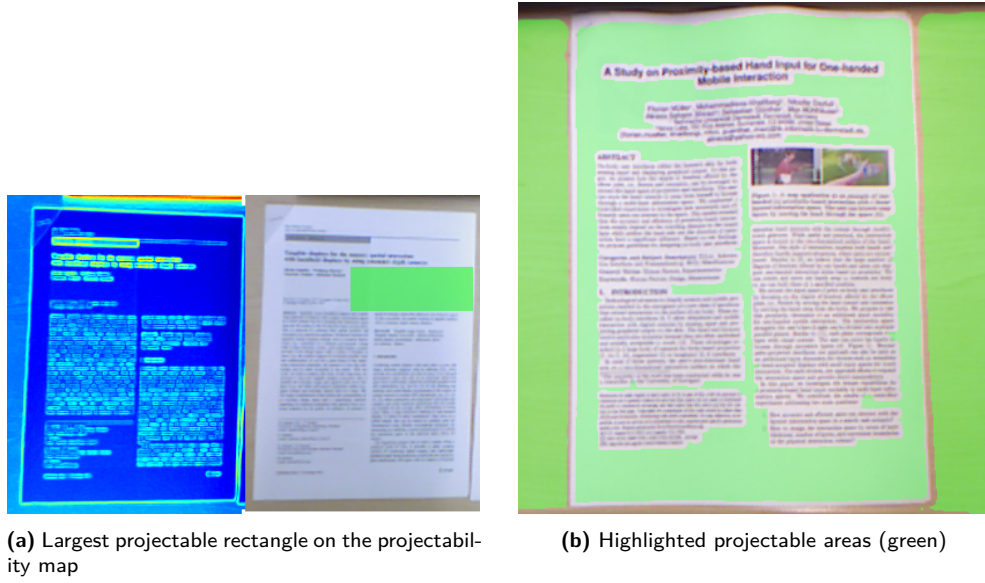


Figure 5.7: Basic strategies for layout generation.

Another possibility is to perform advanced layout based on the original map, like a gradient-based optimization approach in which the object is moved along the gradient of the map and scaled as well as rotated to find an optimal spot projection quality-wise. We will explore this usage in depth in the next chapter.

5.4 System Description and Experience

We implemented the FreeTop concept in the context of the PeriTop hardware setup described in section 2.6. Since FreeTop does not provide any output, it uses only the RGB and depth camera mounted above the table. Hence it is equally usable in the context of the PeriTop and the PeriTop light setup. Software-wise, FreeTop is implemented in C++ using OpenCV [Bradski, 2000] for the image processing.

To be able to provide a first assessment of its performance in a lightweight setup and as a proof of concept, we search the largest available rectangle assessed suitable for projection using the online mode previously described (i.e., no caching, fully processing every frame). We ran FreeTop on a Linux PC with an Intel i5 processor. The OpenCV GPU acceleration was not used, as it is not available for all hardware. Frame time was measured over map computation and rectangle finding across 100 frames.

At full HD resolution, the average frame processing time is about 74 ms

(13.5 fps, independent of scene contents), which is suitable for projectability assessment in low-dynamic scenes (e.g., a desk). For scenes with higher dynamic, a lower resolution can be used, allowing up to 60 fps at 640×480 . Through parallelization or GPU acceleration, the performance can be further improved.

5.5 Summary and Conclusion

The FreeTop approach presented in this chapter provides free spot identification in arbitrary environments. It thereby lays the foundation for projection at suitable free spots on documents that is required by the previously introduced StackTop concept. Hence, it is one of the core building blocks for systems that tightly integrate digital and physical content. By enabling informed projection onto physical objects that takes possible interference into account, it expands available display space. Thereby, FreeTop also contributes to resolving the issue of competition for space between the different objects present in the hybrid work environments described in the introduction of this thesis.

Beyond that, FreeTop can be applied to a multitude of usage scenarios, from general on-surface augmentation using projection to virtual overlays displayed using AR glasses. Contrary to existing approaches, FreeTop does not rely on a single feature but uses a multitude of factors (lightness, smoothness, physical surface, and user preferences) to assess projectability on a continuous scale. This allows flexibly adapting projection to the situation, as it is possible to respect trade-offs (e.g., closeness to the intended position vs. projection quality). Because it is based on a standard depth camera assembly, which is available off-the-shelf and integrated into modern AR glasses, FreeTop can easily be used in both stationary and mobile settings.

Besides supporting the rather basic layout strategies described in this chapter, FreeTop is also the foundation for the more advanced, optimization-based layout approach described in the next chapter.

FlowPut – Embedding Tangibles

As outlined in the introduction of this thesis, tangibles are important means of input on interactive surfaces. However, support for tangibles is currently limited in the sense that their presence is detected on tabletops and input is usually generated from their position and rotation. They currently often lack means of in- and output on their surfaces which causes the tabletop display area around them to be used for tangible-related output. In hybrid settings, this adds to the problems of scarce display space and the tangible-related output also suffers from occlusion problems. In order to overcome these problems, this chapter introduces the FlowPut concept which, in contrast to current state-of-the-art approaches, allows providing input and output based on external touch sensing and projection on nearly arbitrary 3D objects. Further, FlowPut not only provides output on the surfaces of tangibles, but also in their surroundings in an environment-aware manner by means of dynamic layout generation to avoid occlusion.

After an introduction (section 6.1), relevant related work is discussed in section 6.2. The chapter continues with an overview of the FlowPut concept (section 6.3) and then explains the details of object tracking and touch input detection (section 6.4), as well as the layout generation for output (section 6.5). In order to demonstrate the applicability of FlowPut, two example application scenarios are described in section 6.6. Section 6.7 provides insights into the practical implementation. To conclude the chapter, an evaluation (technical as well as a small-scale user study) is presented in section 6.8 and current limitations of FlowPut are discussed (section 6.9). The chapter closes with a summary (section 6.10).

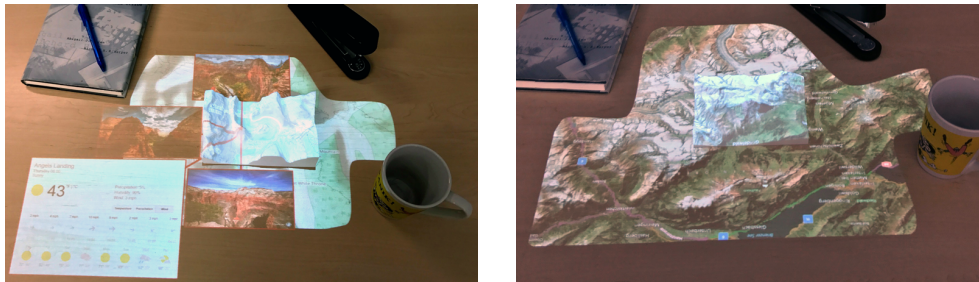


Figure 6.1: The space on and around a tangible is augmented with an environment-aware projection that places free-floating UI elements like images or information widgets (left) or maximizes the space of an UI element (right) based on the environment.

6.1 Introduction

In the previous part of this thesis, we explored the use of on-object visualization using the peripheral display concept to provide awareness for occluded digital documents and foster the integration of digital and physical documents by providing hybrid stacking. All those contributions have been in the field of document integration.

However, as outlined in the introduction of this thesis, there is a second group of physical objects on tabletops: *tangibles*. Tangibles have been proposed [Ishii and Ullmer, 1997] as a response to the lack of haptic feedback in current touchscreen-based user interfaces that provide a more direct kind of manipulation, but still lack an actual haptic dimension. Also, such tangible objects are well suited to carrying three-dimensional information, which is often lost on flat-screen-based 2D interfaces, especially for non-experts (e.g., interpreting a 2D-relief map compared to a 3D tangible of the relief map).

Unlike documents that usually do not have a specific meaning in the context of a user interface, tangibles do have a meaning and are an integral part of the user interface. As such, digital objects related to the tangible are not additional information, as is the case for information about an occluded object (cf. ProjecTop) or digital objects in a stack (cf. StackTop), but an important part of the user interface that has a meaning for the interaction with the systems.

Thus, problems may arise when using tangibles together with other objects on the same surface. In the context of the hybrid setup envisioned in this thesis, they need to fit into an existing ecosystem with foreign “non-tangibles” present on the desk. As a result, the screen space might be limited. Also, tangibles are currently mostly passive objects without input and output capabilities on

their surfaces. Interaction is usually mediated through the aforementioned UI elements displayed around them on the table surface. As a result, they are also prone to the problems of limited display space and occlusion. In order to overcome these problems, FlowPut builds on the FreeTop approach described in the previous chapter to inform the placement of digital objects related to tangibles. By using an optimization-based approach, the objects can be automatically placed to avoid occlusion and provide a high visual quality by avoiding interference.

While in the past, tangibles were often prefabricated objects delivered together with an application, advances in 3D printing have enabled users to produce customized tangibles ad hoc. As a result, the user of a tangible application does not only have to rely on a given set of tangibles but can easily add new tangibles to the set. This poses a new challenge to application developers, since the application then needs to support these user-provided tangibles. Recently, 3D printing technology advanced to a level where it became possible to integrate basic means of in- and output into printed objects. As a result, 3D-printed objects are no longer passive pieces of plastic. This allows users to fabricate interactive 3D-printed tangibles [Schmitz et al., 2015; Willis et al., 2012] on demand. While promising, the current state of the art in this field is still limited and does not yet provide a wide and sophisticated range of input and output techniques: It is, for instance, not yet possible to print an object that has its complete surface covered with a display. Adding to these practical issues, the whole process is very complex, time-consuming, and requires a lot of expertise. Hence, the applicability for non-expert users is currently limited. Further, active electronics, which might be required, need an energy source, making the use of such tangibles even more cumbersome (e.g., one would have to swap batteries or charge them).

In order to overcome all these issues, we propose FlowPut, a framework to provide input and output for many 3D objects (see figure 6.1). To that end, FlowPut tracks 3D objects based on their digital 3D models, augments their surfaces and surroundings using a top-mounted projector, and provides multi-touch input across the surfaces of objects and the table. Further, we contribute environment-aware layout functions for placing UI elements on and around the object that avoid clutter and bad projection quality (e.g., on text documents that are lying around). The 3D models required for object tracking can either be provided from an existing file or be scanned ad hoc, allowing the use of many 3D objects with FlowPut. For developers, FlowPut provides touch

and object movement events and automatically generates environment-aware layouts, consisting of standard UI controls.

In summary, the contributions of this chapter are as follows:

- A point-cloud-based object recognition and tracking for 3D objects providing low latency recognition and real-time tracking.
- A point-cloud-based touch input detection for 3D objects that provides multi-touch not only on the upside of an object, but also on its sides.
- A projection mapping and layout approach allowing environment-aware projected user interfaces by considering optical and physical properties of the projection surface.

6.2 Related Work

Since FlowPut directly builds on FreeTop as one of its core components, the related work discussed in chapter 5 is also relevant for the present chapter. For redundancy reasons, we skip the discussion of these works in this chapter. Please refer to section 5.2 for details.

6.2.1 Touch Input on Surfaces

Touch input has become a popular input modality that is often used for tangible user interfaces. There are several different approaches to provide touch input on and around tangible 3D objects.

The placement of the touch-sensing hardware can be used to divide these approaches into two categories: Touch-sensing hardware can either be embedded into the object (*embedded sensing*) or outside of the object (*external sensing*). Embedded sensing has the advantage that everything is integrated into a single object and there is no or little (e.g., a Bluetooth receiver) external hardware required. The disadvantage is that the objects used need to be manufactured to contain the sensing hardware, which increases cost and complexity. For external sensing, the advantages and disadvantages are exactly the opposite: The advantage is that it can be used with (nearly) arbitrary objects at the cost of requiring external hardware setups.

Embedded Sensing

Touch functionality can be embedded into or attached to tangibles by utilizing embedded sensing. In this case, the sensing hardware is directly attached or even integrated into the tangible itself.

For distinguishing *individual predefined touch points*, this can be done by either using capacitive [Sato et al., 2012] sensing, which works by sensing capacity differences between two sensor electrodes that form a capacitor, or by means of acoustic sensing [Ono et al., 2013], which works similarly but detects changes in the frequency response based on whether an object is touched or not. As a result, these two approaches can deliver information on whether an object is grabbed or not. Also, it is possible to integrate touch-sensitive pipes into objects. Thereby, more distinct touch points can be detected since the pipes denote specific areas on the object surface instead of the object as a whole. To assign touches to the respective points denoted by the pipes, researchers have used capacitive signatures [Savage et al., 2014] or infrared emitters and sensors, either for coarsely distributed sensing points [Willis et al., 2012] or for more high resolution sensing by means of computer vision techniques using an infrared camera [Brockmeyer et al., 2013]. Brockmeyer et al. [2013] even provided input and output via the same set of pipes by adding a projector.

Besides individual points, *continuous 2D touch surfaces* can be built using capacitive touch sensor foils: Research has emerged that allows direct printing of such foils and electronics in general, e.g., via inkjet printers. As a result, custom-shaped touch foils can be generated that support specific buttons [Savage et al., 2012] or multi-purpose sensing areas similar to conventional touch-screen foils [Gong et al., 2014] that detect multiple fingers. While the aforementioned approaches rely on direct capacity sensing, other approaches have been explored as well, like time-domain reflectometry (TDR) that allows retrieving the position of a touch on a single sensing electrode [Kawahara et al., 2013]. Other than that, even cuttable touch sensors that can be cut to shape later have been developed [Olberding et al., 2013], which also have been combined with visual output [Olberding et al., 2014]. To avoid using specifically designed foils, the use of electric field tomography was studied by Zhang et al. [2017]. This allows detecting touch by coating the surface of an object with a conductive layer and sensing changes in the electric field with a comparatively small number of sensing electrodes.

While these approaches are widely used, these approaches often require

manually assembling additional electronics (acoustic sensors, touch electrodes, etc.) or are only applicable to simple, developable surfaces (touch foils for post-assembly). While this is often not an issue, it is impractical for ad-hoc printing interactive tangibles due to the higher effort required and cost of the hardware.

External Sensing

Another stream of research utilizes external sensors, mounted in the environment (such as RGBD cameras), to enable touch input on a large variety of surfaces and objects without the need to alter the tangibles or surfaces themselves.

Historically, the first approaches worked with standard *color cameras*, since depth cameras did not exist yet. Most notably is the DigitalDesk [Wellner, 1991] that uses a conventional camera to detect touch and pen input. But also more recent works like the EnhancedDesk [Koike et al., 2001] rely on similar vision-based approaches. The PlayAnywhere system by Wilson [2005] in contrast uses an infrared-based approach in which the scene is illuminated with infrared light and a stereo camera pair equipped with infrared filters that observes the scene and detects touch input.

With the advent of *depth cameras*, the approaches rapidly shifted towards the use of such cameras, since they provide a simple means to detect touch. Pioneered by Wilson [2010], this depth-camera-based approach was rapidly adopted, and frameworks emerged that provide not only touch input on static surfaces, but also basic object tracking and touch on the surface of objects [Klompaker et al., 2012]. These approaches have been extended from desk use to room-scale use [Xiao et al., 2013]. Also, hybrid approaches using depth and infrared sensing have been explored [Xiao et al., 2016].

Today, depth-camera-based approaches can be considered state-of-the-art for external touch sensing. Since it does not add require any modification of the objects, depth-based sensing is well suited for FlowPut. Generally, depth-based touch detection delivers 3D-touch coordinates and can handle dynamic scenes where objects are moved. Yet, most of the current approaches [Klompaker et al., 2012; Wilson, 2010; Xiao et al., 2016] rely on the concept of background subtraction, which is suitable for static scenes and can be extended to dynamic scenes by continuously updating the background model. However, this approach is often limited, for instance when body parts, present

	Dynamic Scenes	Touch on Upside	Touch on Sides	Coordinates
<i>Embedded Sensing</i>				
Olberding et al. [2013]	●	●	●	2D
Ono et al. [2013]	●	○	○	-
Sato et al. [2012]	●	○	○	-
Savage et al. [2014]	○	●	●	-
Savage et al. [2012]	●	●	●	2D
Brockmeyer et al. [2013]	○	●	○	2D
<i>External Sensing</i>				
Klompmaaker et al. [2012]	●	●	○	2D
Koike et al. [2001]	○	◐	○	2D
Wellner [1991]	●	●	○	2D
Wilson [2010]	○	●	○	2D
Xiao et al. [2013]	○	●	○	2D

Table 6.1: Properties of a representative selection of touch detection approaches. Embedded approaches require sensors to be attached to or into the object. External approaches use sensors in the environment of objects. ● indicates that a requirement is fulfilled, ◐ indicates partial fulfillment, and ○ that a requirement is not fulfilled.

during the update, need to be removed from the background model. Also, touch capabilities are usually limited to surfaces facing towards the camera, neglecting touch on the sides of objects. In order to expand touch sensing to the sides of objects to provide richer interaction capabilities, FlowPut uses a point-cloud-based approach instead of background-subtraction.

Requirements and Summary

In the following, we present a set of requirements for hybrid stacking support derived from reviewing the related work and use them to compare the related work (see table 3.1).

R1 Dynamic Scenes For a wide range of applications, the environment is not static but changes over time (e.g., objects are relocated on a desk while the user is working there). Thus, the system must be able to accommodate changes in the environment in real-time.

R2 Touch on Upside Touch support on the upside surfaces of objects must

be supported, as it can be considered state-of-the-art.

R3 Touch on Sides In order to provide richer interaction capabilities, touch recognition on the sides of objects is also desirable, for instance to detect grasping.

R4 Coordinates Especially in the context of tangibles with non-developable surfaces (i.e., surfaces that cannot be built by cutting a single sheet of paper), 3D coordinates are desirable.

In contrast to prior work (see Table 6.1 for a comparison), we contribute an approach based on point-cloud processing instead of a background model being subtracted to detect touch. Thus, our approach inherently supports dynamic scenes since there is no static background model. The touch points are established by detecting intersections of the point clouds of the objects with the point clouds identified as body parts. The intersection-based approach allows us to expand touch from the upside of objects to the sides of objects as well. This is currently only possible with embedded sensing, which has the downside of requiring additional electronics within the tangible. Further, FlowPut can deliver the 3D coordinates of a touch relative to an object instead of being limited to 2D surface coordinates.

6.2.2 Object Recognition and Tracking

Projection mapping requires an accurately calibrated projector and camera system. The extrinsic and intrinsic parameters of the camera are usually known or can be calculated, for instance, using Zhang’s method [Zhang, 2000]. Calibration parameters for the projector are similarly computed by finding the projected 2D positions of known 3D locations in the physical scene [Bandyopadhyay et al., 2001; Raskar et al., 2001] and then using them to texturize objects via projection in 3D space dynamically.

The object tracking method used for projection mapping must provide high precision and high tracking speed to avoid misalignments between the tracked object and the projected texture. Over time, many approaches have been developed that can be categorized into three categories: *Marker-based* approaches that use some kind of marker attached to the object for identification and tracking, *optical* approaches that are based on cameras, and *depth-based* approaches that use a depth camera and either process the depth data as an image or as point cloud.

Marker-based

Previous approaches used magnetic sensors [Bandyopadhyay et al., 2001; Kondo et al., 2008; Ullmer and Ishii, 1997] to determine position and orientation of objects. However, magnetic sensors work only at short distance ranges. In another direction, fiducial markers to obtain the position and rotation of objects have been widely used. Placing a visual marker also allows identifying documents [Koike et al., 2001; Mitsuhara et al., 2010]. Since they are quite large (e.g., up to 5×5 cm [Koike et al., 2001]), smaller dot-cluster markers [Narita et al., 2017] that are printed using visible ink, or markers that are printed using infrared ink and hence invisible to the user [Kim et al., 2016] were developed.

While easy to use, these approaches require specialized hardware, additional sensors, or markers that need to be attached to the objects. These are often visible to the user and would interfere with projection onto the objects.

Optical

To overcome these issues, researchers have investigated optical methods that use image analysis to recognize objects or documents. This can be done by directly processing the camera image to identify objects like documents [Liao et al., 2010; Wellner, 1993]. However, these approaches are affected by projection on objects since the appearance is changed through the projection, causing the features to no longer match the ones stored in the database. This can be avoided by using infrared cameras in combination with structured-light-based approaches [Cotting and Gross, 2006]. However, these are not able to identify documents and can only detect the presence of objects. Also, they require the objects to be tall enough so that they deform the light pattern.

In summary, these approaches either rely on the visual appearance of an object or cannot identify specific objects. An advantage over the marker-based approaches is that they do not require mounting any markers or otherwise altering the object. However, using the visual appearance might cause further problems when content is projected onto the object or for objects that do not have a strong visual texture.

Depth- and Point-Cloud-based

Like in the field of touch detection, depth-based tracking methods had a considerable impact on object detection as well since they are not intrusive as they rely on the 3D model of the objects and a depth camera. Since depth cameras often use the infrared spectrum, they are immune to interferences from visual projection. The captured depth data can be processed in various ways to detect objects.

For example, to detect single deformable objects, the use of analysis by synthesis has been proposed and used to track flexible sheets of paper [Steimle et al., 2013] for which no static model can be established. When a static model exists, shape templates can be used to train a model for object detection [Kobayashi and Hashimoto, 2014; Tsuboi et al., 2013] or point-cloud-based approaches that directly process the 3D points captured by the camera [Koizumi et al., 2015; Zhou et al., 2016]. Besides detecting specific objects, finding (flat) planes to display content is also common and done using depth cameras, e.g., to provide spatial constancy for augmented reality content [Ens et al., 2015] or to detect cubically shaped objects to display content [Sano et al., 2015].

In order to be able to track objects, an initial pose is required that is then updated during tracking. The initial pose estimate is often acquired by matching Fast point feature histograms (FPFH) [Kobayashi and Hashimoto, 2014; Rusu et al., 2009; Tsuboi et al., 2013], while the pose updates in the tracking phase are computed using the Iterative closest point algorithm (ICP) either alone [Rusu et al., 2009; Tsuboi et al., 2013] or in combination with a particle filter [Kobayashi and Hashimoto, 2014; Koizumi et al., 2015]. However, depth-based methods are computationally very expensive, and the camera and projector also contribute to an additional delay, which has to be compensated by predicting the motion of objects [Bandyopadhyay et al., 2001; Koizumi et al., 2015; Zhou et al., 2016].

While they are the most complex technologically, point-cloud-based approaches have a set of advantages for FlowPut: Just like the optical approaches, they work with unaltered objects. Since depth cameras operate in the infrared spectrum, they are immune to projection. Further, they do not rely on visual features, but the 3D structure of the object. For 3D-printed tangibles, this information is readily available through the 3D model used for printing. For other objects, it can be obtained by scanning the object with a depth camera.

	Object Recognition	Object Tracking	Immune to Texture Changes	Unmodified Objects
<i>Marker</i>				
Bandyopadhyay et al. [2001]	●	●	●	○
Kim et al. [2016]	●	●	●	○
Koike et al. [2001]	●	●	●	○
Mitsuhara et al. [2010]	●	●	●	○
<i>Optical</i>				
Cotting and Gross [2006]	○	◐	●	●
Liao et al. [2010]	○	◐	○	●
Wellner [1991]	●	●	○	●
<i>Depth</i>				
Ens et al. [2015]	○	◐	●	●
Steimle et al. [2013]	○	●	●	●
<i>Point Cloud</i>				
Rusu et al. [2009]	◐	◐	●	●
Tsuboi et al. [2013]	◐	●	●	●

Table 6.2: Properties of the most related object recognition and tracking systems. For object tracking, ● indicates full tracking and ◐ recognition only. For object recognition, ● indicates a specific object, ◐ only the category of an object are identified, and ○ means only presence of objects is detected. For the rest, ● indicates that a requirement is fulfilled, ◐ indicates partial fulfillment, and ○ that a requirement is not fulfilled.

Requirements and Summary

In the following, we present a set of requirements for hybrid stacking support derived from reviewing the related work and use them to compare the related work (see table 3.1).

- R1 Object Recognition** Objects need to be recognized reliably and the recognition needs to be able to distinguish different objects.
- R2 Object Tracking** Once an object is recognized, it must be tracked so that the system can assign object-identifiers (e.g., ids) that are constant over time.

R3 Immune to Texture Changes When use together with projection, the recognition and tracking must not be affected by the projection.

R4 Unmodified Objects Ideally, it is not necessary to physically modify an object to be used together with the system (e.g. by attaching markers).

In contrast to prior work (see Table 6.2 for a comparison), we contribute a method to detect and track objects using depth measurements from a consumer-grade depth camera that provides real-time performance without implementing motion prediction. Since the approach is based on depth data, it is immune to texture changes caused by projection. By comparing the point clouds of the present objects with reference point clouds to recognize objects, the need to modify the object, for instance by attaching markers, is eliminated. Additionally, we apply an inlier-based method of the ICP algorithm, which allows stable tracking of partly occluded objects (e.g., by the user’s hands).

6.3 FlowPut Overview

FlowPut tracks 3D objects, provides touch input on and around them, and projects visual output onto those objects and their surroundings. The projection additionally takes into account the surface structure of the object and the environment by leveraging the FreeTop approach (cf. chapter 5) to avoid projection on unsuitable surfaces (e.g., strongly textured or dark). This functionality is conveniently provided as an API, thereby making it easy to develop full applications requiring interactive tangibles or quickly prototype interactive systems.

To this end, FlowPut requires a hardware setup providing a projector, a depth camera, and, optionally, an IR camera. As a result, it can be directly deployed within the PeriTop environment described in section 2.4. Additionally, FlowPut can also be used on arbitrary surfaces (i.e., conventional desks instead of a tabletop). Since from a general perspective, interactive 3D objects are useful beyond the scope of hybrid work environments (e.g., for interactive board-games), we chose to implement the FlowPut Framework on top of the *PeriTop light* setup (cf. section 2.4.1).

The optional IR camera allows keeping track of the environment without interference from the projected content. Figure 6.2 shows a schematic overview of the minimal setup with the optional IR camera.

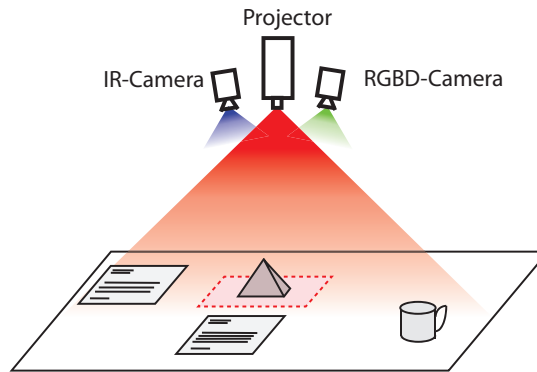


Figure 6.2: Overview of the FlowPut environment consisting of a projector, a RGBD- and an IR-camera mounted above a conventional table. FlowPut tracks known tangible objects, detects occluding (or other) objects, and projects digital content, e.g., contextual information, on and around the tangibles.

We identified the following requirements to ensure broad applicability of FlowPut:

1. *Real-time performance*

The object recognition, tracking, and layout generation should be performed in real-time, as required for interactive systems.

2. *Stable and occlusion-resilient object recognition and tracking*

Predefined objects that are recognized once should be tracked in a stable manner to avoid flickering and jumping of visualizations. The framework should be able to track multiple objects simultaneously and tolerate cluttered environments, since the target setting is office desks and tables. As the user's hands or other body parts may cover the objects during movement, tracking should be resilient to partially occluded objects.

3. *Unmodified Objects*

FlowPut should be able to work with arbitrary existing objects or simple 3D-printed objects without the need to modify them (e.g., by attaching markers or integrating electronic parts).

4. *High touch accuracy and resolution*

For an interactive system, touch input should be possible. To allow fine-grained interaction on tangibles, a high resolution of the touch detection approach is required. Further, the number of false and not-recognized touches should be minimal to ensure a positive user experience.

5. *Occlusion and interference aware UI*

The UI layout of the framework should respect the physical environment of

the objects and avoid interference and occlusion by dynamically clipping or relocating the UI elements with respect to a dynamically changing environment.

6.4 Object Tracking and Touch Input Detection

Since FlowPut aims at augmenting tangibles by means of projection without the need to modify the objects with markers or similar user-visible additions, object tracking and touch input detection rely on depth data only. The reason for this decision is that projection would influence image-based methods, since it changes the textures of the objects that usually serve as a reference for feature detection.

The depth data captured by the camera is internally processed as a 3D point cloud with each input depth pixel resulting in a point in the point cloud.

6.4.1 Input and Reference Model Preprocessing

Capturing a complete scene leads to large point clouds. With increasing camera resolutions in the future, they may even further increase in size. At the same time, processing large point clouds is a time-consuming task, since most algorithms (e.g., the iterative closest point algorithm) have a complexity of at least $O(n)$ or even $O(n^2)$ relative to the number of points processed. In order to still achieve real-time performance, reducing the number of points while at the same time retaining the necessary details to ensure the stable recognition of objects is an important challenge.

Model preprocessing

FlowPut aims at accepting models from different sources, e.g., 3D scans, models for 3D printers, etc. These models can significantly differ in their resolutions – for instance, a very highly detailed CAD model for 3D printing. Since the resolution of the model might even be higher than what the printer can print or the depth camera can sense, the amount of data would cause a severe performance penalty without any advantages. Hence, the models are preprocessed to adjust the resolution to fit the parameters of the system. Since the reference models of the objects to be tracked are static, the runtime time requirements can be reduced by preprocessing the models at or even before

the application startup. During this process, the resolution of the point cloud of the objects is downsampled to a resolution of a point-to-point distance equivalent to 2 mm in the real world in order to match the resolution of the point cloud data from the depth camera.

Plane removal

A large part of points captured by the camera belongs to the table surface. Since the table surface is a flat plane, it can be represented by a simple plane equation ($z = 0$) and hence, the points belonging to the table surface do not have any value for the following processing steps. Thus, in the plane removal step, all points belonging to the surface plane of the table are removed from the input point cloud captured by the depth camera (see figure 6.3 left versus center, where the table plane is removed). However, the noise floor of the depth camera causes the points representing the surface of the table to be randomly distributed above and below the actual surface of the table by up to 1 cm. The majority of these points can be removed by applying a height threshold (e.g., $\text{abs}(z) < 0.25 \text{ cm}$). It must be noted that the threshold values mentioned here mainly result from the current state of technology (e.g., noise of the depth measurements provided by the camera) and not a general limitation of the FlowPut approach.

By doing so, most of the remaining points of the surface of the table are removed. However, at the same time, the lower parts of all objects on the table are also lost. Thus, this imposes a minimum height on the objects on the table that should be tracked so that enough points are above the threshold and not removed in this step. Nevertheless, the remaining points from the point clouds of the objects are still sufficient for robust pose estimation. As mentioned, the process removes most table surface points, but not all. In order to remove the remaining ones as well, FlowPut uses a statistical outlier test that tests if a point has at least five neighbors within a small radius of 0.75 cm. In case there are not enough neighboring points, it is likely a residue of the thresholding or an object too small for detection and tracking. Thus, the point is then removed.

Clustering

Once the input point cloud is cleaned up, the remaining cloud points need to be separated into per-object clusters for each object present in the scene.

Based on the clustering, it is decided which step in the processing pipeline is applied next (see figure 6.3 center versus right, where the point cloud is split into four separate clusters):

For tracked objects that were present in the last camera frame, the last positions and rotations of the objects are already known from previous frames. Thus, FlowPut can extract the points belonging to these objects using the known positions and dimensions of the tracked objects. A small margin is added to these values to capture changes in position or orientation between frames. Since the cluster type (e.g., hand or object) is already known from earlier frames, the cluster can be directly passed to the corresponding tracking method.

A special case is the act of touching a tracked object. In this case, the body cluster and the cluster of the touched object merge. However, by leveraging the known pose of the tracked physical object, it is still possible to correctly segment the point cloud by first extracting the points belonging to the object based on the known pose and assigning the remaining points to the body cluster.

The remaining points must then belong to unknown objects. In order to separate them, they are clustered using a Euclidean clustering algorithm with a 5 cm cluster-to-cluster distance. These clusters are then split into two categories: the user’s body and (currently) unknown objects that might need to be tracked if belonging to an object known to FlowPut. To decide between the two categories, FlowPut checks whether the cluster intersects with the field of view of the camera. This is based on the assumption that a user’s arm needs to reach into the tracked workspace area from its outside, as the user is located in front of the desk. In case there is an intersection, the clusters are classified as body-party independent of their actual form. As a result, they are passed to the touch detection stage for further processing. All other clusters are passed to the object recognition stage.

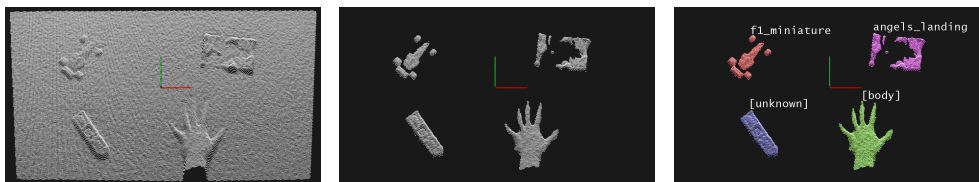


Figure 6.3: Processing pipeline of the point cloud data: raw (left), preprocessed (center), and clustered/tracked (right).

As a result of treating unknown clusters this way, an object is only recognized as an object if it is placed on the table surface and the user removes his hands. Otherwise, it will still be classified as a body-part, since it is connected to the hand-cluster that intersects with the field of view.

6.4.2 Object Recognition

In order to recognize objects through their point-cloud, a database of known objects is required. This database contains reference point clouds of all objects the system should be able to recognize. The respective reference point clouds need to be provided to the framework by the user or application using FlowPut. This data is then used to identify objects from unknown clusters and estimate their 3D positions and orientations as described in the following.

First, the reference point cloud from the database needs to be aligned with the input cluster to be able to check them for similarity. In previous work [Kobayashi and Hashimoto, 2014; Rusu et al., 2009; Tsuboi et al., 2013], 3D features and matching algorithms have been employed to find a proper alignment. However, these methods are computationally very expensive while not being robust and accurate enough for the noisy input data provided by current depth cameras.

In order to overcome these issues, FlowPut uses principal component analysis (PCA) [Jolliffe, 2011] to align the isolated unknown clusters with the reference models that are registered as trackable. The PCA yields the three principal axes for an object in the order of their variance from the cluster. Also, we know that the object is placed on the table, which is a flat surface, and we know which side of the point cloud faces upwards. This leads to a fixed z-axis that points upwards from the table-surface. Thereby, we can reduce complexity by projecting both point clouds into the x-y plane and comparing only the largest principal axis to get a rotation angle around the upward-facing z-axis. Since PCA only yields an axis, but no direction of that axis, FlowPut checks both complementary possibilities for the rotation angle (α as well as $\alpha + 180^\circ$). To compensate the translation between the two clouds, they are both moved such that their respective centers of mass are at the origin of the coordinate system $((0, 0, 0)^T)$.

The computed principal axis of the point cloud provides a rotation estimate for the object. For objects that are rectangular with different side lengths, the principal axis is easily computed. However, for objects with a quadratic

bounding box, the principal axes might be indistinguishable. While in the first case there are only two possible orientations to be checked, in the latter there are four, spaced by 90° . As long as the surface structure of an object is not symmetric, the correct rotation can be found by matching the actual points in the next step. In cases where the structure is symmetric, the rotation is inevitably ambiguous. As a result, for completely round or cylindric objects, the rotation around the main axis cannot be determined.

Now that the alignment is known and reduced to at most four possibilities, the quality of the match is computed using a fitness score. The score is based on the accumulated distance of each point in the point cloud of the reference model to its nearest neighbor in the aligned unknown cluster. The resulting sum is then normalized by the number of points in the cloud, yielding the average point-to-point distance, which is independent of the cloud size of the model. If the fitness score is below a certain threshold (e.g., average point mismatch less than 2 mm), indicating a good alignment, the unknown cluster is identified as a trackable object of the type of the current reference point cloud. Based on this initial recognition, the object tracking algorithm is initiated using the pose estimate derived from the PCA alignment. As explained in the previous section, this information is directly used in future camera frames so that the recognition process will not be executed again.

In order to further increase the system performance, the fitness score computation, which is linearly dependent on the number of points in the point cloud of the model, the point clouds are downsampled to a resolution of one point per centimeter. This step results in point cloud sizes of about 100-200 points depending on the physical size of the model. This reduction strongly decreases the computation time, while still maintaining enough expressiveness to determine the matching quality.

Clusters that could not be matched to a known object from the database in this step are temporarily labeled as unknown objects. Thus, they are processed as unclassified again in the next frame. The reason is that the cluster might belong to a known object; however, due to shadowing caused by the user's hands or other objects, it might be unrecognizable at the moment.

6.4.3 Object Tracking

Once a cluster is recognized as an object, it is subsequently tracked by FlowPut. This is performed using the iterative closest point algorithm (ICP) [Zhang,

1994] that is initialized with the last known orientation. ICP then minimizes the matching distance of two point clouds by iteratively transforming them based on their respective point-to-point nearest neighbor matches. Thereby, the transformation parameters for rotation and translation are iteratively approximated.

One of the basic assumptions of the ICP algorithm is that there is only a small misalignment between both object clouds between two consecutive input frames. Otherwise, it might fail at converging towards the correct transformation. In order to avoid this, changes in orientation and position need to be minimized between two consecutive ICP tracking iterations. Since the lower bound for the time intervals between the iterations is the input frame rate of the camera, we aimed at reducing the computation time for each frame so that FlowPut can process each frame. Thereby, changes in orientation and position are kept to a minimum. In order to reduce the computation time, FlowPut uses the aforementioned downsampled and extracted point clouds.

The ICP algorithm uses a similar score as the previously described fitness score. However, in order to account for possible occlusion (e.g., by the user's hands or other objects being moved over the object), point-to-point distances that are larger than 2.5 cm are ignored. Thereby, points that are currently occluded, and would otherwise be matched with points that are further away, do not reduce the fitness score. As long as the remaining points are still expressive enough, ICP can compute a good alignment.

Due to the nature of iteratively approaching the pose, the pose returned by the ICP algorithm is not stable across processed frames and may slightly vary, even if there is no actual change in the scene. This variation, in turn, causes the projection in later stages to jitter accordingly. To avoid this effect, which is unappealing and disturbing to the user's eye, the pose is smoothed over a short time to provide a stable pose. A negative side effect of such a filtering approach is that it causes a delay when the object is actually moved. To compensate for this and minimize latency, FlowPut employs a movement detection approach that turns off the aforementioned smoothing of the pose during actual movement.

6.4.4 Touch Input Detection

Besides tracking tangibles and providing output on their surfaces, FlowPut is also able to detect touch input on the table surface and on the surfaces of

tracked objects.

Leveraging the point-cloud-based approach described before, FlowPut can as well detect touch input on the sides of object, as follows. This is in contrast to background-subtraction-based approaches that only provide touch on surfaces facing towards the camera. To detect touch points, the clusters that were classified as body clusters in the preprocessing step are used. Instead of further segmenting the body clouds in order to detect actual fingers, which would be computationally complex, touch detection is performed using the whole body cluster.

Since a cluster is a body cluster when it intersects the field of view of the camera, as described before, regardless of its shape, objects held in hand function as extensions of the body. As a positive result, touch is not only possible using fingers but also using a pen or paintbrush held in the hand. However, when the user holds an object in order to place it on the table, a undesired false touch event is detected if the object touches the table or another object. In order to minimize false touch inputs, the touch points detected are filtered depending on their size, rejecting points that are larger than a finger. Since the typical maximum fingertip size is about 20 mm in diameter, it is safe to reject touch points of more than 25mm in diameter. Thereby, it is still possible to recognize touches by fingers or pens or even touches with corners of larger objects (which can be seen as a targeted pointing action), while larger touch areas that occur when placing an object on the desk are rejected.

To avoid generating touch events for objects that are held in hand to be moved around or removed from the scene, tracked objects are removed from the tracking process if they are lifted more than 10cm from the table surface. Thereby, they become again a part of the body cluster until they are put down again and recognized as objects. As before, all thresholds are configurable and the values provided here are for illustrative purposes and have been found suitable in our practical experiments for our hardware setup.

The whole process is implemented as follows: For each body cluster identified, the first step is to compute the distance between each point in the body cluster and the surfaces of the table or objects. For the table, this distance is given by using the plane equation representing the table surface. For objects, the point cloud of the model is transformed according to its last known pose, and then a nearest-neighbor search [Knuth, 1998] is performed for each point of the body cluster. Thereby, most of the body cluster can be ignored, and only points close to the table or an object are further considered since they indicate

touched areas. Such areas can be caused by one or multiple fingers or even larger parts of the body touching the table or an object. Next, touched areas are generated by clustering the previously identified points to have clusters with a minimum distance of 1 cm. If the cluster has a size smaller or larger than a fingertip (e.g., a whole hand resting on the table or noise) the touch area is discarded. Based on a resolution of 2 mm to 2.5 mm, the typical fingertip, sized 16 mm to 20 mm [Dandekar et al., 2003], has a point cloud consisting of 5 to 50 points.

Since the touch points are directly generated from the body clusters, each touch point is directly linked to its respective cluster. As a result, one could expand FlowPut with user tracking capabilities, which link the body clusters to a specific user around the table. This would, in turn, enable multi-touch for multiple users by assigning a specific user to the touch points. By tracking the detected touch points over time, the well-known touch-down, touch-move, and touch-up events are generated. Further, the tracking is used again to smooth the position of the touch events over time by using exponential filtering. Thereby the accuracy and robustness are increased by removing the jitter and compensate for single noisy measurements.

To ease application development, FlowPut allows specifying particular areas on an object which trigger “named” touch events (so-called *touch regions*). These allow the application to be model agnostic by, for example, waiting for a touch event for the “roof” region rather than matching the general touch events to the geometric location of the roof. The so-called *touch regions* act as a virtual button and can be easily defined using a simple tool and provide more semantical meaning to the applications than just a simple 3D coordinate. Internally, touch regions are stored as subsurfaces of the point cloud of the object.

6.5 Environment-aware UI Layout

While people are working, tables are usually cluttered with a wide variety of different objects with different textures (e.g., paper documents, pens, or notebooks). A naive projection system without an awareness of its environment would produce an irritating user interface that projects over such objects in an uninformed manner, resulting in poor user experience and usability (e.g., digital text projected over a printed text often renders both unreadable, as depicted in figure 6.5).

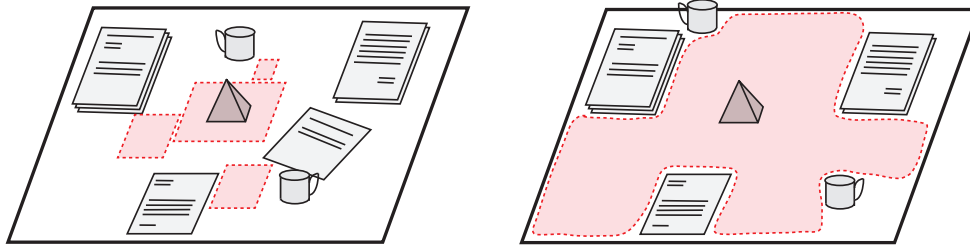


Figure 6.4: Layout for UI elements based on environmental constraints (left) and maximum space layout (right). Projection is shown in red.

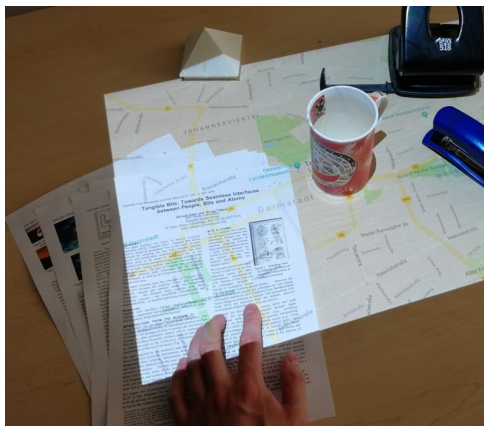


Figure 6.5: Interference between projection and the content of a physical document.

In order to reflect the presence of different objects and textures, while also providing a sophisticated output on and around a tangible object, layout techniques that consider the environment are required to place UI elements. As illustrated in figure 6.4, we propose two techniques that utilize the surrounding of objects for visual output: (1) placing UI elements based on environmental constraints, and (2) projecting as much visual content in the surroundings of the tangible ob-

ject as possible without jeopardizing projection quality. Variant 1 (see figure 6.4 left) is suitable for tangibles that use projection to be texturized and to display a small context area as well as widgets displaying related content (e.g., a 3D-printed machine control panel that has its texture projected on. To be able to see the location of the control on the machine, a small part of the machine is displayed around the tangible. The widgets could be additional UI controls to control the application or display additional information). Variant 2 (see figure 6.4 right) is suitable to display as much context as possible for a tangible, e.g., a 3D-printed mountain relief plus the larger context in the form of a map to make it easier to understand the geographic location.

The environment-aware UI layout techniques described in this chapter rely on the FreeTop per-pixel projectability measure described in the previous chapter. Figure 6.6 shows the projectability map together with a set of flowing elements (gray) that are laid out using the projectability map. While the elements are all associated with the same target position (the black dot within the red rectangle that represents the physical object they are associated to),

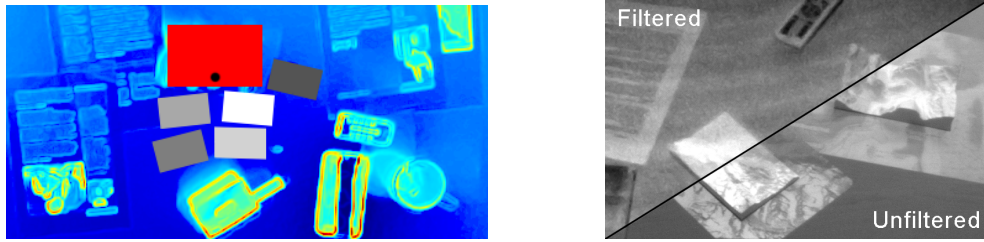


Figure 6.6: Visualization of the element [gray] placement on top of a projectability map [blue=good score, red=bad]. The target position of the elements is the black dot on the physical object [red box] (left). Image captured by the IR camera with and without IR filter (right). The projected content is invisible in the filtered image.

FlowPut automatically places them at areas with good projectability in the vicinity of the target location.

6.5.1 Interference

Obviously, the (visual) projectability measure is influenced by the projection. For example, when considering a completely white, even surface, the projectability would allow projection anywhere. However, if the framework decided to project at a place P , the next projectability calculation would suggest avoiding projection at P , as the camera picks up the projected image. The projection would then be moved to another place, and the process would start over. To avoid this, we used an offline computation approach that computes the projectability of, for instance, an object or the environment before the projection starts and caches the map for future use (e.g., projection of text onto objects). This approach is very suitable for environments where only tracked objects are moved and the textures remain static.

6.5.2 Projectability

Since we target dynamic scenes as well, using cached maps would be a severe limitation. Therefore, we extended the caching approach as follows: Since the framework knows where it projects, the cached projectability map can be updated regularly in areas where no projection occurs, filling the projection areas with the cached information. To account for texture changes in areas with projection, we used an additional IR-camera equipped with a low-pass filter to filter out visible light, i.e., the projection (see Figure 6.6 right). In this way, the areas with projection are monitored, and if significant changes in (IR) texture are detected, the projection can be very shortly disabled to update

the map in this area. Then, based on the updated map, the projection can be re-enabled and, if necessary, moved to another place.

Using this approach, the projection needs to be disabled to update the map. To not completely disable the projection, one could synchronize the camera with the projector and generate a short gap within one of the colors-phases of the projector (in the case of a DLP projector) as done by Junuzovic et al. [2012]. However, this is rather complex and requires modifying the projector. Therefore, we propose another approach: Projectors with laser light sources, which are becoming more and more popular, use three (R,G,B) very specific wavelengths. These wavelengths can be filtered for the camera by using narrow-band notch filters so that the camera can continuously monitor the area in the visible spectrum without picking up the projection at all. While this eliminates the need for the additional IR camera, such filters are expensive and make the setup more complex (i.e., the wavelength of the filters must match the wavelength used by the projector).

6.5.3 Placement Based on Environmental Constraints

For the placement of additional related information, the layout process should consider the projection quality of the target surface and a set of other constraints as well. For instance, labels should be as close as possible to the object they label and be oriented towards the users to improve readability while being displayed on a surface with good projection quality.

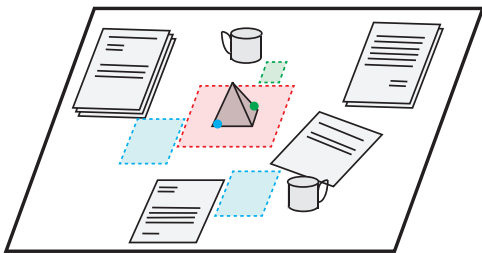


Figure 6.7: Example of an optimized layout.

Since these constraints often contradict each other (e.g., a big projectable space far away versus a less projectable small space nearby), we propose using an optimization-based approach to generate a UI layout that computes an optimal place for all UI elements based on a set of environmental constraints (see Figure 6.4 left

and Figure 6.6 left). Figure 6.7 shows an example of such a layout. The digital objects (blue and green rectangles) are augmentations to the pyramid-shaped tangible in the center and belong to the marked spots on the pyramid (dots colored similarly to the objects). Using the optimized layout, their positions are automatically assigned and are updated when the environment changes

(e.g., objects are moved).

The elements to be placed are characterized by their 2D position, size (width and height), and orientation: $v = (x, y, s, \alpha)$. Width and height are represented by a scaling factor s to maintain the aspect ratio of the element. We formalized the optimization as a minimization problem, since our goal was to minimize the total costs C_{Total} for the placement of all elements \mathcal{E} with each element $e \in \mathcal{E}$ having a set of costs \mathcal{C}_e where each cost $c(v)$ is weighted with an individual factor $w_{e,c}$:

$$\min C_{\text{Total}}(v) = \sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}_e} w_{e,c} c(v)$$

This formulation allows for an individually weighted set of constraints for each element based on their individual properties (e.g., readability is more crucial for a text element than it is for an image element that tolerates more interference). For single elements, it is sufficient to optimize v by only considering the projection quality and the element-local constraints (e.g., minimum size). In the case of multiple elements, it is also important to ensure that the elements do not overlap. Therefore, additional constraints specifying that the pairwise intersections of all elements must be empty (i.e., no element is overlapping with any other) were added to the set. To avoid unwanted flickering, we added a motion filter that only repositions the elements if their new locations largely differ from the last positions.

Constraints

The optimization currently considers the following constraints but can be easily extended with additional ones.

Projection quality: To ensure a good projection quality, the surface of the table is assessed by the framework based on the projectability map P , which contains a continuous projection quality value for each pixel. The projection quality for an element is then computed by averaging the individual pixel values $P_{x,y}$ from the map across the area of the element e given its current location, rotation, and scale:

$$c_P = \frac{1}{\text{width} \cdot \text{height}} \sum_{(x,y) \in e} P_{x,y}$$

Proximity to the desired location: The position cost is modeled as the Euclidean distance between the desired location p_d and the actual location

$p = (x, y)$ of the element:

$$c_{pos} = \sqrt{(p_d - p) \cdot (p_d - p)^T}$$

Orientation towards the user: Since the projected content is ideally oriented towards the user who requested the element to be displayed, the angle cost is modeled as the absolute difference between the angle facing towards the user α_d and the angle α of the element itself:

$$c_{angle} = \text{abs}(\alpha_d - \alpha)$$

Besides the cost for deviations from α_d , a hard constraint on the maximum rotation is enforced (i.e. $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$). This is necessary to avoid rotating elements by 180 degrees and to ensure readability of text elements, since too much rotation often renders text unreadable.

Scale: Depending on the content, an element may be scaled to fit into a projectable area. This can be achieved using a scale constraint that is modeled as the absolute difference between the original scale of the element s_d and the actual scale s of the element:

$$c_{scale} = \text{abs}(s_d - s)$$

As for the angle, scaling is a constraint on the minimum and maximum scale (i.e., $s_{\min} \leq s \leq s_{\max}$). In most cases, s_{\max} will be equal to s_d with both being one (the element should be displayed at its intended size and not larger).

Normalization

The constraints are not consistent in their value range, i.e., the scaling factor is a value between 0 and 1, the orientation angle ranges from 0 to 2π , and the projection quality ranges from 0 to 255. If the cost values were directly summed up, the projection quality, for instance, would have a much higher impact than a deviation in scale or rotation and therefore would always outweigh them. To consider the different ranges and allow for individually weighting the different metrics of any element, we normalized the cost values c to the interval $[0, 1]$ and then weighted them using element-specific factors. If necessary, this can be extended using more complex (e.g., non-linear) normalization functions. Initially, all weights are 1, so every factor is weighted identically. However, users may change the weights if desired.

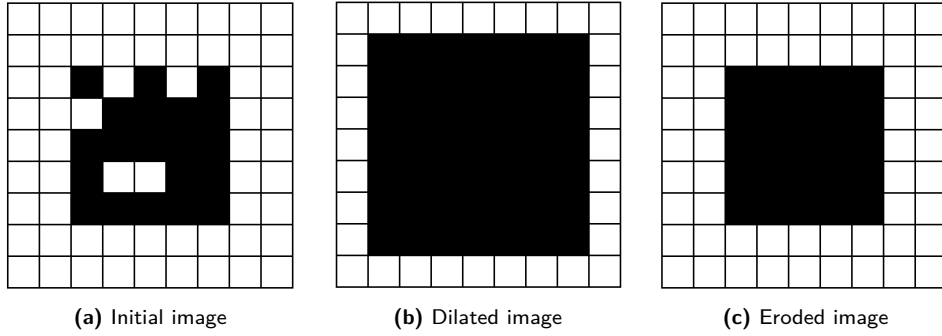


Figure 6.8: Example of a Dilate/Erode iteration. The initial image (left) is Dilated (center) and then Eroded (right). Thereby, the holes in the original square are closed.

6.5.4 Maximum Space Layout

Besides the need to place information related to the tangible in its surrounding, there are situations in which the tangible needs to be viewed in its context. For instance, in the case of a tangible relief map, the framework can not only texture the 3D tangible itself, but also display the geographic map information on the table around the tangible (see Figure 6.1 right). However, such context information is fixed in its position relative to the tangible and thus cannot be placed with our optimization technique. Also, the information displayed should be as large as possible. While one can display this in a naive way by just projecting everywhere within the reach of the projector, this is problematic on cluttered tables, because the projection heavily interferes with other physical objects on the table.

Therefore, we propose using the projection quality measurement to layout the elements around obstacles using a flood fill [Burger and Burge, 2009b] approach: Based on the projectability map, the algorithm starts at the centerpoint of the tangible and stepwise expands the area from there to all surrounding pixels of the map that are below a predefined threshold. In order to close tiny holes and smoothe the contour, two iterations of dilate/erode [Burger and Burge, 2009a] are performed. Doing this expands the contour from the previous step by two pixels in every direction, and then shrinks it again. Thereby, in the dilate step, holes that are smaller than 4 pixels are closed and in the erode step, the contour is brought back to its original size (see figure 6.8 for an example). To improve the stability of the detected occlusion regions at the border areas and prevent flickering of the contour, we performed additional smoothing and filtering operations. Since the contour generated by the flood-fill approach is not smooth, we approximated a low complexity polygon of the contour and

transformed the linear line segments into a continuous Bézier curve [Guillot, 2008] to improve the visual quality.

The UI element that is to be maximized in space is then clipped according to the connected projectable area around the tangible denoted by the Bézier curve.

6.6 Applicability

FlowPut is designed as an integrated solution for providing interactive environment-aware applications that are based on a large interactive surface combined with tangibles that are themselves touch-enabled and provide output. In order to demonstrate the applicability of FlowPut in such scenarios, we consider two example use-cases in the following to which FlowPut is particularly well suited: (1) interactive tangible 3D displays and (2) interactive tangible 3D proxies. We will explain the value of FlowPut for these two categories through two interactive sample applications, namely, (1) a tangible relief map and (2) a tangible car proxy (see Figure 6.9).

6.6.1 Interactive Tangible 3D Displays

Leveraging the surface of everyday objects as an interactive information display has become common and frameworks are emerging to facilitate this process (e.g., the toolkit support by Hardy and Alexander [2012]). By extending these interactive surfaces beyond flat two-dimensional (table) surfaces to 3D tangibles, FlowPut furthers this vision. As a demonstration for this category, we implemented an interactive map relief based on a 3D-printed tangible object. The tangible represents the geographic structure of the area of interest and is then visually augmented with the corresponding texture via FlowPut. Thereby, an otherwise flat map becomes three-dimensional, which fosters spatial understanding (see figure 6.9 left). As an example of interactivity in such scenarios, users can search for photos taken at various locations by touching a point on the tangible map. The corresponding photos are then arranged around the tangible using the optimization-based layout approach. At the same time, the projected map leverages the maximum space layout approach to use the available space around the tangible optimally (see figure 6.1 left).

From a technical perspective, implementing such an application using FlowPut is straightforward: after registering the 3D model of the tangible within

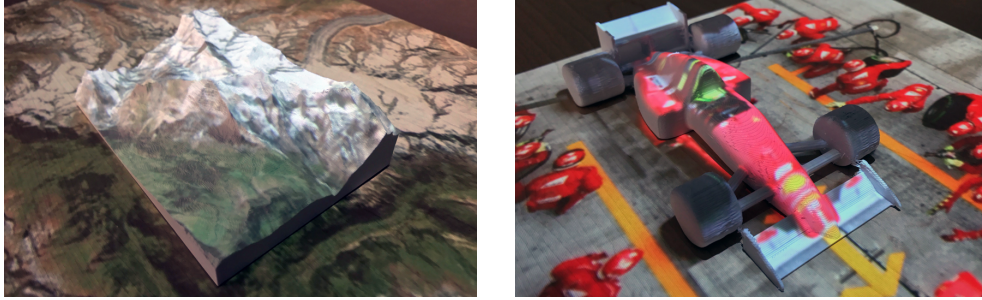


Figure 6.9: Two interactive example applications: a tangible relief map (left) used as a tangible 3D display, and a tangible car (right) used as a 3D tangible proxy.

FlowPut, any existing UI content or control can be easily bound to the position and orientation of the tangible. In the registration step, the reference 3D model is added to the database of known objects and an identifier is assigned that FlowPut uses to reference the object (e.g., when transmitting position updates). Moreover, FlowPut automatically clips the UI elements according to the maximum space layout. Detected touches on the tangible and table are automatically redirected to the respective UI elements. In our sample, touches on the tangible are used to retrieve images for the respective geographic locations and register them within the FlowPut framework to be optimized. The position and rotation of each image are continuously and automatically adjusted to an adequate space on the table.

In summary, an existing standard map application could be easily transferred to FlowPut by interchanging the layout-controlling UI components (e.g., grids) with FlowPut and adding the code to initialize FlowPut (i.e., adding the respective models). Of course, the tangibles with their corresponding 3D model need to be available. However, for 3D-printed tangibles, this is usually the case, since the model is needed for the printing process and this model can be directly used within FlowPut.

6.6.2 Interactive Tangible 3D Proxies

Configuring physical objects using a computer is common nowadays, be it CAD-models (e.g., for later 3D printing) or physical products that can be customized before order (e.g., via a configurator for cars or furniture). For expert users, such configurators or CAD software tools are conveniently usable after going through a steep learning curve. However, for novice users or occasional users, this is often not feasible. FlowPut can be used to provide an interactive configuration experience by enabling users to interact with a tangible proxy

object and change surface properties (e.g., the outside color of a car could be changed by directly touching the surface and choosing a color) or customize the physical structure (e.g., by tapping on parts and enabling/disabling them). Changes are then visualized instantly on the proxy via projection and, at the same time, in the configuration software that is connected to FlowPut. Thereby, the configuration experience becomes more direct and has an additional haptic component that is missing in screen-based systems. Further, FlowPut is not limited to a single proxy within the workspace but can work with multiple proxies in parallel. As a result, it can also be used in planning scenarios as well, e.g., the placement of machines in a factory or city-planning. Each machine or building is then represented by a tangible proxy that can be individually configured and placed.

As an example application for this category, we implemented an interactive tangible car proxy (see figure 6.9). Using FlowPut, users can specify the color of the car and the surface materials, or control the interior of a car via touching the specific part of the tangible proxy. The updated specifications are then either displayed in-place (e.g., the color of the car is directly projected onto the surface) or within a separate widget when the property is related to the interior and hence not directly visible on the tangible (e.g., interior views showing the color of the seats), enabling a more direct manipulation than conventional touch- or mouse-based interfaces. Furthermore, by touching a part of the car, additional information is shown at the most suitable place on the table using the constraint-based layout algorithm (see figure 6.1 left).

New proxies can be added by registering their 3D models within the framework and binding a dedicated UI element to the position of the proxy. The named touch regions facilitate binding the new model to the existing UI elements: By naming the corresponding touch regions in the same way on all proxies, the respective functions can be triggered without modifying the application code. For instance, every car has tires that should invoke the “select tire” dialog, so one could generate four touch regions named “tire left front”, “tire left back”, etc. that work independently of the actual object.

6.7 Implementation

Contrary to ProjecTop and StackTop concepts in the field of stationary hybrid workspace, we use the simplified *PeriTop light* hardware setup for FlowPut. The idea is that once the integrated projector-camera lamp (cf.

figure 2.5 in section 2.4, page 35) is available, FlowPut can be used in an ad-hoc manner, even in areas where no high-resolution tabletop is available. Hence, our lab implementation is based on a standard full HD projector (1080p) mounted 2.35 m above the table together with a Kinect V2 depth and color camera for projectability assessment and an additional IR camera (uEye UI-548xSE-M with an IR low-pass filter) to detect texture changes. The cameras are mounted about 95 cm above the table (see 6.10). Our demo objects are 3D-printed with a standard BCN3D Sigma 3D printer and regular white Verbatim PLA filament. The respective 3D models are preprocessed using a Meshlab filter script, which normalizes the triangle mesh by applying a uniform mesh resampling, re-computing the vertex normals, and exporting the simplified mesh into the .PLY format.

The FlowPut framework is written in C++ using Qt. Object recognition, object tracking, and touch detection process the depth data from the Kinect in a point cloud data structure. Therefore, we use the plane removal, noise removal, Euclidean clustering, nearest-neighbor-searches, iterative closest point algorithm, and principal component analysis methods provided by the point cloud library (PCL) [Rusu and Cousins, 2011]. The calibration from Kinect to the projector, computation of the projectability maps, and computation of the maximum space layouts all use OpenCV [Bradski, 2000]. To optimize the placement of 2D UI elements according to the projectability maps, we use the non-linear optimization library NLOpt and the gradient-free local optimization using the principal-axis method. In our current setup, the user is assumed to be standing on one side of the table and the rotation of elements is constrained according to this assumption (i.e., all elements are rotated to be readable from one side of the table). The software implementation, however, allows defining different target orientations for each element to reflect multiple user positions, e.g.,



Figure 6.10: Our lab setup.

provided by a user tracking system. Communication with other applications is based on Google’s Protocol Buffers for serialization and TCP as the network protocol.

Applications for FlowPut can be written in any language or run on separate machines, as the FlowPut API is accessible via ProtoBuf messages sent over TCP. Thereby, it is possible to use any desired UI toolkit. For instance, our example applications are built in C# using WPF.

6.8 Evaluation

To assess the applicability and to ensure coverage of a broad range of scenarios, we evaluate and discuss each of the four previously defined requirements in the following.

6.8.1 Real-time Performance

<i>Method:</i>	Technical Analysis	The user experience heavily relies on the application being responsive enough to directly detect and respond to any user input (e.g., touch, object, or occluder movement). Therefore, real-time performance is a key element for the
<i>Data Gathering:</i>	Logging	
<i>Variables:</i>	Processing times	

Table 6.3: Evaluation overview.

FlowPut framework. Each frame has to be completely processed before the next frame arrives (maximal 33 ms at 30 frames per second).

We measured the computation times for the individual processing steps during a normal use case scenario with up to three tracked objects and three unknown objects at a time along with touch interaction and up to 12 individual free-flowing UI elements. The measurements were conducted on a desktop computer running Windows 10 with 8 GiB RAM and a 3.6 GHz Intel i5 Processor.

For our setup, the preprocessing took about 15.6ms ($\sigma = 6.1$ ms), which was mainly caused by the expensive noise filtering and clustering processes. After the initial preprocessing, all of the following processing steps could run in parallel and therefore their times do not add up to each other. Object detection required 1.2 ms ($\sigma = 0.4$ ms) and object tracking 4.3ms ($\sigma = 1.9$ ms),

which was, however, influenced by the size of the model. Touch detection and tracking are only evaluated if the user’s hands are present in the scene and took 0.3 ms ($\sigma = 0.6$ ms).

Optimizing the arrangement of the UI elements and computing the maximum space layout are not performed on a frame-by-frame basis and hence do not interfere with the critical path. Instead, they are constantly running and updating the layout in the background with the currently available object positions and projectability maps. The time taken for the layout optimization is mainly dependent on the number of elements. For the eight UI elements in our experiment, the optimization took 12 ms ($\sigma = 5.4$ ms). Computing the maximum space layout for two objects took 8.4 ms ($\sigma = 4.5$ ms) in our experiment.

This led to a maximum processing time of 19.9 ms ($\sigma = 6.4$ ms) per frame, which is below the 33 ms limit required to process all frames. However, the Kinect and projector used added about 155 ms of latency, largely outnumbering the latency of the framework and leading to a visible delay when moving objects at higher speeds. This is, however, not a limitation of FlowPut, and is likely to be resolved with future hardware generations that provide lower latency.

6.8.2 Stable and Occlusion Resilient Recognition and Tracking

For our example applications, we utilized models between 15.8 cm \times 10.0 cm \times 4.8 cm and 14.3 cm \times 9 cm \times 3 cm in size. However, our tests revealed that models down to 5 cm \times 5 cm \times 2 cm can be detected. The minimum height of 2 cm is due to the fact that most of the points of the object need to be above the table surface with respect to the noise floor of the Kinect of ± 5 mm.

<i>Method:</i>	Quantitative
<i>Participants:</i>	5
<i>Duration:</i>	ca. 15 min
<i>Data Gathering:</i>	Logging
<i>Variables</i>	
<i>Independent:</i>	Speed, occlusion
<i>Dependent:</i>	Tracking status

Table 6.4: Study design overview.

To evaluate the occlusion resilience of the tracking, we manually occluded an object with cardboard after the initial recognition. We tested occlusions of 0% to 90% relative to the surface of the object in 10% steps. We recruited five participants, all male and aged between 29 and 35. While partly occluded, the object was moved by the participant along a predefined path with a length of

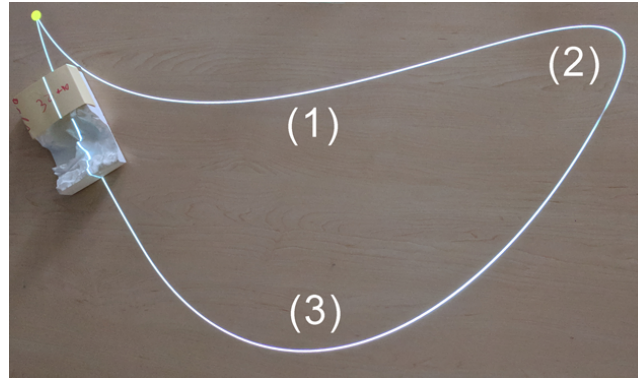


Figure 6.11: General setup with a 40% occluded object to evaluate the performance of our tracking approach.

215 cm across the table. Since the tracking is also dependent on the speed of the object, participants moved the object at five different speeds (5 to 25 cm/s in 5 cm/s steps). To ensure constant and correct speed, participants had to move the object while following a projected yellow dot that moved along the path (see figure 6.11 for the setup and the path). The path was chosen to contain a nearly straight section (1), a 180-degree turn (2), and a slightly sloped section (3). The object was moved along the path once by each participant for each combination of occlusion and speed. This led to a total of five recordings for each combination. We counted the number of successful trials. A trial was successful if the tracking was never lost during the trial and the detected pose did not differ more than 3 mm and 5 degrees along any axis from the predefined path.

The results are shown in Figure 6.12. As one would expect, the tracking becomes more unstable with both increasing occlusion and increasing speed. At 25 cm/s, tracking was always lost while moving the object, even without occluding the object. With 90% occlusion, tracking was lost either directly upon occlusion or after a few centimeters of moving. The loss at 90% (or 15.8 cm² of visible area) occlusion is consistent with the minimum size of the object of 25 cm² (5 cm × 5 cm). At 5cm/s, the slowest speed, tracking was only lost once at 80% occlusion. For a more practical speed of 10 cm/s, tracking was stable for up to 50% occlusion. At even higher speeds the success rate drops further: At 15 cm/s, we observed a 100% success rate up to 10% occlusion and only 20% at 20% occlusion. At 20 cm/s, we observed 60% success without occlusion and 40% with 10% occlusion.

In practice, this limitation does not pose a severe restriction because the occlusion of an object is usually caused by the user's hands while moving the

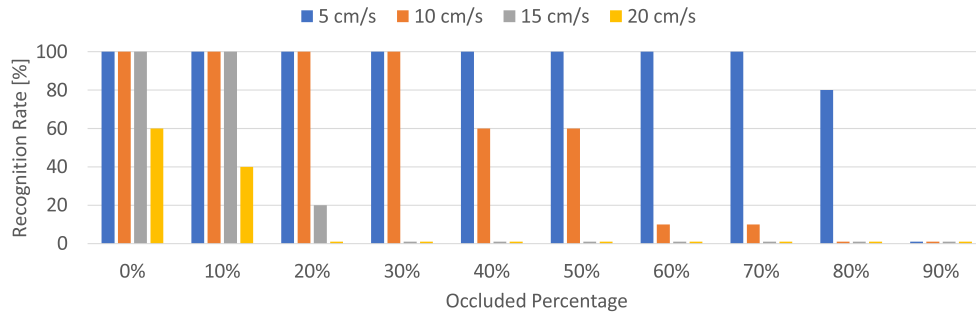


Figure 6.12: Results of the evaluation of the performance of our tracking approach.

tangible, so the occlusion is only temporary. The tracking is immediately restored after the user removes his hand and the object is recognized again by the framework. Hence, this situation only leads to short outages of the projection during fast movements. This is especially prominent with small objects where the hand can occlude almost the whole surface while the object is grabbed. However, this is a general problem of all vision-based approaches, since a marker or the object itself needs to be visible to the camera.

In 76.47% of the cases, tracking was lost at the narrow curve during the 180-degree turn (see (2) in Figure 6.11), i.e., when a fast rotation is combined with linear translation. Therefore, the results could further improve for non-rotating move-only interactions. In general, we noted that the occlusion performance of the tracking is also related to the surface geometry of the object: areas without distinguishing surface features are not as important for the tracking as are highly structured areas.

6.8.3 Touch Accuracy

In order to evaluate the touch detection accuracy, we performed a small-scale evaluation with the same five participants as the occlusion study. To that end, we selected five different representative surface geometries, namely two vertical planes to evaluate touch on object sides (one parallel to the field of view of the camera and one perpendicular), a convex spot standing out of a surface, a concave

<i>Method:</i>	Quantitative
<i>Participants:</i>	5
<i>Duration:</i>	ca. 10 min
<i>Data Gathering:</i>	Logging
<i>Variables</i>	
<i>Independent:</i>	Touch location
<i>Dependent:</i>	Measured coordinates

Table 6.5: Study design overview.

	Touch Plane	σ_x	σ_y	σ_z
Side (Frontal)	x-z	2.95	0.01	4.73
Side (Side)	y-z	0.21	2.83	4.82
Convex	x-y	2.13	1.77	1.39
Concave	x-y	5.25	2.08	1.63
Flat	x-y	2.54	2.08	0.00

Table 6.6: Standard deviations of the reported touch coordinates for different surface geometries in mm.

area, and finally, as a reference, a spot on the flat surface of the table.

The spots were marked on the model and repeatedly touched, each three times by each participant, leading to a total of 15 samples per point. A projected blue dot indicated a successful (i.e., detected by the framework) touch on the target area. Apart from two failed touches inside the concave area, all others were successful. The touch coordinates reported by FlowPut were recorded. We computed the standard deviation as a measure of repetition accuracy from the values. The results are shown in Table 6.6. The surface of the table is in the x-y plane, and the z-axis is perpendicular to the table plane in an upward direction.

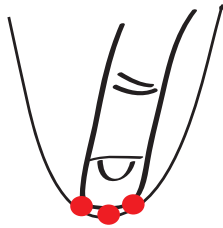


Figure 6.13: Ambiguity of touch points in concave areas.

Touch on concave surfaces yields the worst results due to the ambiguity of the intersection, as there are multiple possible touch points from the perspective of the camera (as depicted in figure 6.13, multiple points of contact (red dots) can be established), from which the center one detected by the camera is picked. This might not necessarily be the exact point the user intended to touch, but a nearby point instead. However, a standard deviation of 5.25 mm along the x-axis is still within the size of a fingertip (cf. [Dandekar et al., 2003]).

For flat and convex surfaces, the results improved further, with only about 2 mm standard deviation on the x- and y-axes and about 1.5 mm on the z-axis (see Table 6.6 for the exact values). Because the camera observes an area of about 1 m in width and 80 cm in height at a resolution of 512×424 pixels, the 2 mm standard deviation is equivalent to 1 pixel in the raw depth image, which is an acceptable performance.

For touch recognition on the side surfaces, there is no huge difference between the two different orientations of the sides. In both cases, the axis perpendicular

to the orientation of the side (e.g., the y-axis for the frontal side) has a very low standard deviation, since its value is directly derived from the tracking data of the model. For the variable axes, we observed a slightly higher variance for the x- (or y, respectively) axis than with a flat or convex area. The z-axis has the highest standard deviation, but it is still within the typical size of a fingertip.

6.8.4 Occlusion and interference aware UI

By utilizing projectability maps for our UI layout optimization, FlowPut can adapt the UI to the physical environment. Since this is an automated process which constantly adjusts the UI layout, we investigated how users perceive such a dynamic UI compared to a static one.

<i>Method:</i>	Qualitative
<i>Participants:</i>	6
<i>Duration:</i>	ca. 20 min
<i>Data Gathering:</i>	Semistructured interview
<i>Conditions:</i>	Flowput, no FlowPut

Table 6.7: Study design overview.

We conducted a user study to assess the impact of an optimization-based layout approach on user experience. To that end, we compared two conditions (static and dynamic) for photo browsing using our tangible relief map. In both conditions, participants were asked to find a specific photo from a set of 10 photos bound to landmarks on the map, which were marked with a red dot. By touching the red dot on the map, participants could toggle the visibility of a photo of a landmark. In the static condition, we displayed photos in a standard list of photos on the right side next to the map. In the dynamic condition, we used FlowPut to automatically layout each photo at a spot close to the landmark. In both cases, the photos were connected to the landmark using a red line. To provide a more realistic environment, we also added common objects to the scene (keyboard, pen, another tangible, and a coffee mug).

There were 6 participants (all male, aged between 27 and 34). We chose a within-subject design. For each task, participants were given time to familiarize themselves with the system. Each session lasted about 20 minutes, and participants were asked to think aloud. We conducted semi-structured interviews after each condition to collect additional feedback specifically regarding the layout aspects.

In the static condition, all participants complained that after placing the

tangible relief map, the projection collided with other physical objects. All but one participant started by moving the physical objects away to have a free working area. While P6 found the presentation with the list structured (since the photos were always displayed at the same place), P2, P3, and P5 complained that the relation between landmarks and photos was often unclear, despite the connecting lines (P5: “*Lines intersect other landmarks*”, P3: “*The lines were sometimes crossing*”). P4 noted that “*the many photos are overextending*”.

In contrast, in the dynamic condition, we observed that all participants opened more photos at the same time and were amazed by the automatic layout. They found it to be better arranged and tidier (P3: “*It’s good that the photos are always close to their location on the map*”, P6: “*It’s nice that the photos automatically avoid obstacles*”, P5: “*Following the lines is much easier now*”). However, P1, P2, and P4 also criticized that there was no way to intervene, e.g., by pinning an object to its current position or manually moving it away (P4: “*It would be interesting to manually move the photo as well*”). Furthermore, participants were occasionally irritated by photos moving due to camera noise. To improve this, position changes could be filtered even more to ignore movements below a certain threshold, thus preventing “*shaking*” (P2) of the projected content.

6.9 Limitations

Despite the strengths of FlowPut, it currently has limitations that must be considered before usage. Some of these limitations are caused by the way FlowPut is implemented and works, while others, especially the restrictions regarding object size and touch accuracy, are caused by hardware limitations and are not inherent to the FlowPut system.

6.9.1 Minimum Object Size

Currently, tangibles should be at least 2 cm high above the table to be tracked correctly. There are also limitations for the overall size of a tangible: it needs to be at least 5 cm by 5 cm. As FlowPut is mainly targeted at tangibles with sufficient space for top-projection, this restriction is usually not an issue. However, there are cases in which users might want to track multiple smaller tangibles (e.g., playing pieces for an augmented board game).

The limitations of the minimum height and size of an object are not inherent

to the algorithms employed by FlowPut, but rather are the result of the noise of the depth camera (e.g., for our setup, camera noise is about ± 5 mm at a distance of 1 m) and overall resolution. Additionally, the flying-pixel effect further influences the precision of the measurements negatively [Gottfried et al., 2014]. Future depth cameras are likely to mitigate this issue due to reduced noise and increased resolution.

6.9.2 Touch Point Accuracy

The location accuracy of the touch points depends on the surface geometry of the object. Touchpoints on convex regions of the surface (e.g., the top of a mountain) result in more accurate location estimates compared to concave or flat regions (e.g., a valley in a tangible relief map). As we select the closest surface point to the fingertip as the touch location, there is usually a unique point for convex regions. However, for concave regions, there are often multiple closest points and hence multiple possible touch locations. While we use the center of the candidate cluster, a lower noise and higher resolution depth camera would increase performance. Future work could also explore more advanced algorithms to disambiguate touch points (e.g., by taking the user’s viewing perspective into account).

6.9.3 Body Part Classification

To determine body clusters within the point cloud data, FlowPut uses a simple heuristic that defines a cluster as belonging to a body if it hovers above the table surface, starts at the edge of the camera field of view, and had not previously been classified as another object. As a result, a new object moved into the tracking area is classified as a body part until it is placed on the surface and no longer touched. Thus, it is not possible to directly interact with a new object without moving the hand away after placing it. Future work should investigate methods to separate the user’s hand from an object held in the hand to mitigate this issue. This would also allow the user to interact with the object while holding it above the table.

6.9.4 Automated Element Placement

FlowPut uses the FreeTop projectability map as a basis for layout generation. As a result, the user mask provided by FreeTop allows the user to mask areas

that are subsequently omitted for projection by FlowPut. However, besides this coarse mean of control, there is currently no way for users to influence the placement of elements in a more fine-grained way. Based on the user feedback in the evaluation, future work should investigate ways for users to actively specify the desired placement in addition to the location computed by FlowPut. For instance, users may move or fixate the location of an element with a three-finger gesture.

6.10 Summary and Conclusion

In this chapter, we presented FlowPut, a framework providing interactivity for tangible objects by visually augmenting them and their surroundings using environment-aware projection and by providing multi-touch input across the surfaces of the tangibles and the surrounding surface of the table. FlowPut aims at the concurrent use of tangible objects and other objects, like office equipment or documents, on the same surface.

Considering the challenges outlined in the motivation of this thesis, the FlowPut concept contributes to a solution to the problem of the competition for space and provides a seamless integration of tangibles in hybrid environments.

To realize this, FlowPut provides environment-aware layout mechanisms to minimize interference and collisions between the projected digital content and the physical environment. As part of the FlowPut concept, a real-time detection and tracking approach for partly-occluded 3D objects based on their 3D models, a multi-touch detection for the sides of objects, and a set of environment-aware UI layout techniques are contributed.

The FlowPut concept was evaluated and validated through a set of first user studies that indicated a good overall performance. Further, the applicability of FlowPut to practical scenarios as well as its limitations were discussed.

With the integration of tangibles through FlowPut, the set of contributions made in this thesis for stationary workplaces is completed. In the following chapter, we will examine mobile work practices and provide a solution for remote-access to stationary hybrid workplaces.

Part IV

Mobile Workspaces

PiraTop – Mobility for Hybrid Workspaces

Spatial organization has been an important aspect of productivity in physical work environments, as outlined in the motivation of this thesis. With the advent of personal computers, the concept of spatial organization has been brought to the digital domain and then, with the advent of tabletop computers, to hybrid work environments where digital and physical documents coexist on a single surface. However, mobile work has become increasingly widespread nowadays, with people working not only in immobile stationary offices but also in trains, at customer sites, etc. While it is easy to take digital documents with oneself on a mobile device, the physical part of the hybrid work environment as well as its layout is lost when doing so. Thereby, a part of the information as well as the spatial structure of this information is lost in mobile working scenarios. However, as mentioned above, the spatial layout is an important component of productivity, and its loss may have a negative impact on productivity.

This chapter presents PiraTop, a concept for providing mobility for the whole hybrid work environment using augmented reality glasses. After the introduction (section 7.1), a review of relevant related work is presented in section 7.2. Section 7.3 provides a brief overview of the design process, which is followed by an overview of the whole concept as well as the results of an initial focus-group session presented in section 7.4, which informs the design of PiraTop. Based on the results of the focus group, the concepts for the stationary workspace part (section 7.5) as well as the mobile part (section 7.6) are introduced in detail. Following a short description of the implementation in section 7.7, section 7.8 presents a user study comparing different visualization techniques for stacks in the mobile AR-based user interface. Section 7.9 then presents an evaluation in the form of a user study of the mobile interface as a whole. Based on the results of the study, section 7.10 derives a set of design implications for future systems from the results of the user studies. The chapter

concludes with a summary in section 7.11.

7.1 Introduction

All of the previous contributions in this thesis were concerned primarily with the integration of physical and digital documents as well as tangibles in a mainly stationary setting. However, mobile work at various locations has become increasingly popular and widely accepted by employers. People are no longer working only at stationary offices with fixed personal workplaces but also from their home office or while traveling (e.g., while being on a train or from a customer site). Mobile work in this sense has been shown to have several advantages, like increased employee satisfaction [Gajendran and Harrison, 2007], better work-life balance [Hill et al., 2003], and lower costs for the employer [Ruth and Chaudhry, 2008]. However, working remotely poses an additional set of challenges like setting up an appropriate environment with access to corporate resources, which leads to a slower adoption rate [Ruth and Chaudhry, 2008]. One issue to this is the still prevalent use of paper. While it is easy to take the digital workspace with oneself using a tablet or notebook, it is cumbersome to transport printed documents, and doing so destroys the spatial layout.

Despite a large amount of supporting systems for stationary work, the field of home and mobile offices is still only coarsely covered in research and practice. For instance, research has emerged to address assistance in these working scenarios. To allow the mobile augmentation of physical documents, pico-projector-based solutions have been developed [Harrison et al., 2011; Mistry et al., 2009]. While these systems allow using digitally augmented documents not only in stationary but also in mobile scenarios, they do not provide access to the whole work environment with its spatial layout and all documents. The spatial continuity of work environments [Ens et al., 2015] has also been explored using head-worn AR systems. The system developed by Ens et al. allows the augmentation of physical workplaces with digital content (e.g., calendars, etc.) that retains its relative spatial layout when switching the physical location. However, mobility of the physical workplace with its documents and layout is not considered.

Going beyond the current state of the art, we suggest to not only augment some elements within a (stationary) office setting but to make the whole desk “mobile”. We propose PiraTop, a novel concept for virtual ac-

cess to physical work environments on the go. PiraTop not only allows the user to carry documents around beyond fixed stationary setups but also allows the user to virtually access his whole work environment wherever he is working by means of augmented reality. Thereby, a user can easily access a document that is lying on his office desk during a meeting or while working at a remote place. Further, PiraTop provides not only “view-only” access to the physical desk but also bidirectional access to the work environment: The user can actually interact with documents on the stationary desk through PiraTop, e.g., by moving or digitally annotating them. While possible for digital and physical documents, moving physical documents requires an additional manual step when returning to the desk, as described later in section 7.5. In order to enable remote interaction with the stationary workplace, PiraTop provides the user a three-dimensional view of his work environment to rely on the spatial clues provided by his personal desk layout. Leveraging the spatial layout enables easy and efficient access to and interaction with the documents stored there. Figure 7.1 shows an exemplary remote desk as visualized by our prototypical implementation (deliberately using a puristic design).



Figure 7.1: An example of an augmented reality view of PiraTop.

More concretely, the contributions in this chapter are threefold: We first describe an integrated system concept consisting of visualization and interaction techniques for mobile remote workplace access. Second, we present the results of two user studies. The first evaluates the developed visualization techniques quantitatively, while the second evaluates the mobile workplace access concept qualitatively. Last, we conclude with a set of design implications for the further development of augmented-reality-based remote access for workplaces. In the next section, we will start with reviewing the relevant related work.

7.2 Related Work

The field of (remote) office work is a well-known area of research in both psychology and computer science. As a result, a large body of research is relevant to our work presented here, of which we will discuss the most pertinent contributions in the following sections. The works were selected to be relevant

to the Pira_{Top} concept and describe aspects that are transferrable to our concept. We grouped the related work into five areas: work practices, spatial organization on tables, augmented desktops, mobile systems, and AR-based systems. The latter category is generally a subcategory of mobile systems, but since AR is especially relevant to Pira_{Top}, we decided to discuss it in a separate section.

7.2.1 Augmented Desktops

Since Pira_{Top} provides mobility for an otherwise stationary work environment based on an augmented desktop system, the related work on augmented desktops presented in chapter 2 is also relevant for the present chapter. For redundancy reasons, an in-depth discussion is omitted here. Please refer to section 2.2.1 for details.

While the approaches for augmented desktops presented in section 2.2 are well suited for stationary work – e.g., it is comparably easy to mount a bright, daylight-capable projector and a tracking camera statically over an office desk – these systems are not lightweight enough to be temporarily carried around to be used in mobile scenarios. Also, not every workplace one might encounter is equipped with such a fixed projector-camera setup. Even if a top-projection setup is available, projection cannot correctly visualize the third dimension, which is important for stacks or orientation within the environment through physical landmarks (e.g., the monitor). While projection-based augmentation is widely used in related work (see the discussion in section 2.2), it still has some inherent drawbacks (e.g., occlusion by the users’ body parts or being impacted by strong light). Further, one might encounter surfaces that are dark or strongly textured, which cannot be easily projected on. Especially in mobile scenarios, there might not be a suitable surface at all, even if the whole projection system is portable. As a result, we use projection only in the stationary part of Pira_{Top} and use augmented reality for the mobile part. Thereby, we can provide a spatially complete visualization at the stationary workspace as well as when working mobile.

7.2.2 Spatial Organization

As shown in the related work discussion in chapter 2, the ability to spatially organize objects is beneficial for productivity in general because of the user’s spatial memory. While the concept of spatial organization was already applied

to document management (e.g., by Robertson et al. [1998]), where it proved to be useful, we did not find any application to mobile knowledge work, especially remote access to documents, yet. For a detailed discussion of related work, refer to section 2.2.2. PiraTop aims at changing this by providing an integrated system leveraging spatial memory in mobile work.

7.2.3 Shift in Work Practices

Especially in the domain of knowledge work, the amount of telework has increased [Ruth and Chaudhry, 2008]. The term “telework” usually denotes work arrangements in which the employee is only working part of his work time at the office of his employer and conducts a substantial part of his work at remote locations like his home, at the customer’s premises, or even while traveling [Fonner and Roloff, 2010; Gajendran and Harrison, 2007]. While research does not entirely agree on the consequences of this trend [Ruth and Chaudhry, 2008], aspects such as lower costs [Bailey and Kurland, 2002; Ruth and Chaudhry, 2008], lower turnover [Gajendran and Harrison, 2007], higher job satisfaction [Gajendran and Harrison, 2007], and improved work-life balance [Fonner and Roloff, 2010; Hill et al., 2003] are considered positive outcomes of this trend. In larger companies, this results in substantial cost savings. For instance, a study at IBM revealed that half of their employees conduct telework, resulting in a cost reduction of about USD 50 million in real estate [Ruth and Chaudhry, 2008]. At the same time, the productivity of their employees on telework increased by 10–20 percent [Ruth and Chaudhry, 2008]. Since people prefer to use paper along with digital solutions [Koike et al., 2001; Luff et al., 1992]; many documents are still printed to have physical copies [Sellen and Harper, 2003]. Given the often considerable number of documents used, it is impractical to always carry them around when teleworking. Usually, one has decided upfront which documents to take, which might lead to documents being missing. Also, when removed from the desk to be taken along, the spatial information coded in the desk layout is lost.

Eldridge et al. [2000] conducted a study to analyze the behavior of workers when taking paper documents to meetings. On one side, they found out that people are more willing to carry paper documents with them to meetings, the closer the meeting is to the person’s office. Also, they observed that people, as a result, often failed to have important or required documents with them for the meeting. The reasons they found for this are various: unwillingness to take bulky documents just in case they were required, unanticipated requests from

others during the meeting, or relying quite optimistically on someone else at the meeting to bring the document along. Summarizing these results, it can be said that although paper itself is quite mobile and flexible in small quantities, mobility for larger amounts of paper is not very affordable in daily routine. Adding to this, the study revealed that people could not always accurately predict their document needs upfront, making the selection of documents to carry along difficult. On the other side, the study also revealed reasons in favor of taking paper to meetings. For instance, people often take documents to meetings to annotate them because they think they might need them or because the document itself is the subject of the meeting.

Summary

Telework has been shown by related work to be a beneficial work arrangement that has a multitude of positive effects for the employee as well as the employer, e.g., better work-life balance and increased satisfaction and productivity. At the same time, it has been shown that access to information while working in mobile settings outside of the primary workplace is still difficult, especially because of the still large use of paper. As a result, there is a demand for the mobile use of or access to paper-based documents that are printed, for instance, for convenience reasons. At the same time, it is not practical to carry all of them around while working mobile or even to meetings. Since one must choose a subset, there is a chance that one is lacking a document. PiraTop addresses these issues by providing remote access to the user's physical document collection.

7.2.4 Mobile and AR-based Systems

Research has emerged that explores the mobile use of digital technology to facilitate access to information. We group these systems in two categories, namely, mobile systems and AR-based systems. Since modern AR systems are mobile as well, the latter group is actually a subset of the first one. Since PiraTop focuses on AR, we chose to cover AR-based systems more explicitly in a separate section.

Mobile Systems

With the increasing miniaturization of information-technology devices, it became possible to develop more and more sophisticated systems for working with digital content while being mobile. The systems developed can be roughly split into four categories.

Mobile Systems for Desktops The first one aims at mobile devices used on stationary desks. Examples in this category are the Bonfire system [Kane et al., 2009], the portable system for anywhere interactions [Sukaviriya et al., 2004], or the PlayAnywhere system [Wilson, 2005]. The goal of all these systems is to provide a tabletop-style interactive surface nearly anywhere the user wants to work. The devices themselves are constructed to be lightweight so that a user can carry them around in a case, but in order to use them, a stationary desk is required. Also, while the systems are mobile in the sense that they can be carried around, their actual use is limited again to stationary settings (i.e., they cannot be carried around while operating). Additionally, they require a power socket to operate.

Handheld Systems The second category aims at genuinely mobile use. One direction of research employs handheld devices, which serve as a kind of lamp used to digitally illuminate an area and show projected content on the illuminated surface. Interaction using a handheld projector [Beardsley et al., 2005], interacting with dynamically defined information spaces using a handheld projector and a pen [Cao and Balakrishnan, 2006], multi-user interaction using handheld projectors [Cao et al., 2007] or the RFIG lamp [Raskar et al., 2006] are examples in this category. The devices employed are meant to be carried around by the user, which enables highly mobile use. On the other hand, they need one hand of the user to carry the device and limit the user's interaction with the projection. When projecting onto a wall, one cannot easily walk to the wall and interact with the projection since the projection distance would be too small. Hence, electronic pens are proposed as remote pointing devices. The LightBeam system [Huber et al., 2012] solves this to some extent as it can be either carried around by the user like a flashlight or put on a desk and serve as a stationary device. Interaction is possible by manipulating the illuminated objects via a depth camera. However, visualization and interaction is limited to a single document or object that is controlled through the illuminated physical object. As a result of the flashlight-like concept, there is no complete overview

of a whole work environment possible. Further, the concept is not aimed at preserving the spatial layout.

Wearable Systems Pioneered by Steve Mann [Mann, 2000; Mann et al., 1999], the technological development has brought up body-worn mobile systems like Omnitouch [Harrison et al., 2011] or the sixth-sense system [Mistry et al., 2009], which do not require the user to carry the device in his or her hands. As a result, both hands are free to perform input gestures; e.g., the hand can be augmented with projection to serve as an ad hoc touch keyboard by the Omnitouch system, or a nearby wall can be used as an interactive screen (sixth sense).

While being pioneering work, the systems above target the mobile access of digital content and not the mobile access to physical paperwork, e.g., printed documents stored in the user’s stationary office.

Document Management As outlined before, mobile work requires access to information (and people) virtually anywhere and anytime [Perry et al., 2001]. The studies by Perry et al. have characterized some of the particular features and problems of mobile work: unfamiliar environments, unanticipated document behavior, difficulties in accessing remote documents, and the frequency of face-to-face interactions with colleagues, in which paper documents play a key role (cf. Luff et al. [1992]). The findings from their studies informed the design of a new system, called Satchel [Lamming et al., 2000; Perry et al., 2001]. While also being a mobile transportable system, the Satchel system is specifically targeting document management. To avoid having to print and carry around documents, the Satchel system allows document tokens to be carried on a mobile device to enable mobility for remote documents. As remote work venues are often unfamiliar and potentially stressful since the structure is unknown (e.g., where to find a printer, fax, etc.), Satchel aims at mitigating these problems as well using service discovery. Satchel allows to remotely browse directories, systems, etc. to access files by means of tokens and enables the user to print them by querying the Satchel system for a local (“Satchel enabled”) printer.

While there was a multitude of systems developed for mobile knowledge access, none of them take care of the physical work environment of a user. They all do a good task at transferring the digital documents into the mobile world, providing ubiquitous access to them and some even to equipment like printers in the user’s environment. However, none of them targeted the physical work

environment of the user or even the spatial layout a user employs on his desk.

AR-Based Systems

While being around for years, augmented reality recently had a strong boost in popularity. This is mainly due to the technological advances over the past years, allowing small and lightweight augmented reality glasses to be built. They now operate in a stand-alone manner without being tethered to a stationary computer, a prerequisite for true mobile use. Also, the widespread use of smartphones allows users to employ augmented reality by viewing the world through the smartphone, where the virtual part is overlaid on the camera image.

The primary use of augmented reality systems currently is games and the augmentation of labels or annotations. A prominent example for the latter field are remote-assistance scenarios where the remote operator can annotate the field of view of a field worker. A widespread practice is the placement of objects [Oda et al., 2015], where augmented reality is used to denote the position where the field worker should put a real-world object. Other work aims at placing labels and annotations in a fixed manner so that the labels stay fixed in the physical world and do not move with the camera [Cidota et al., 2016; Gauglitz et al., 2014a,b], for instance, to mark specific landmarks in a real-world scenery. Also, in the field of machine control, work emerged to explore ways to guide users when operating complex machinery [Kuzuoka, 1992]. One common aspect of all these works is the spatial referencing in the real world, which is transferred virtually to a remote site. However, all objects considered in these works are not “self-contained” digital artifacts but rather annotations that only have meaning in their current position and often context (e.g., a label “Turn” next to a valve in a machine). As a result, they are not carried around or accessed in other places since they would not have any meaning there.

In another direction, work emerged to use augmented reality in personal use cases where no remote operator is in place but the user can place digital objects within his environment to work with them. At the same time, mobility is provided for the environment by reproducing the spatial layout at other places as closely as possible. The most prominent is the spatial constancy concept by Ens et al. [2015]. It allows, for instance, hanging up a digital calendar behind the desk on the wall. While this is generally an easy thing

to do with augmented reality, Ens et al. went one step further by allowing the transformation of the digital augmentation using an ego-centric coordinate system. This allows the digital objects to not only be shown at the place where they are originally placed (e.g., the work desk at the office) but also transform the layout and show the very same objects also at the workplace in the user's home office. Thereby, the spatial layout, which is essential for the user, is preserved (e.g., the calendar is always in front of the user on the wall), while supporting different environments. However, the system is limited to the placement of individual objects, which can be rearranged to some extent without influencing the meaning to the user.

PiraTop now expands on this by making the whole workspace mobile, not only the specific digital items within the workspace. To the best of our knowledge, this is the first attempt at using augmented reality for remote workspace access.

Requirements and Summary

In the following, we present a set of requirements for hybrid stacking support and use them to compare the related work (see table 3.1).

- R1 **Digital and physical documents** Hybrid workspaces consist of digital and physical documents. As a result, both categories must be accessible during mobile work.
- R2 **Mobile ad hoc use** Mobile access can be needed while working at a remote workplace (e.g., in shared office spaces or meeting rooms) or while on the go (e.g., in trains where no table is available). Both environments should be supported to provide access anywhere.
- R3 **Spatial layout** As outlined in previous chapters, spatial organization of documents is an important aspect of productivity. As a result, remote-access solutions should convey the spatial layout of documents to facilitate document retrieval.
- R4 **Two-way communication** Workers might need to not only retrieve information but also loop back changes (e.g., move a document into the focus area to be reminded of it when returning to the workplace). As a result, two-way communication between the mobile and stationary system parts should be provided.

	Digital and physical documents	Mobile ad hoc use	Spatial layout	Two-way communication	Information exchange
<i>Mobile Systems for Desktops</i>					
Kane et al. [2009]	○	◐	◐	○	○
Sukaviriya et al. [2004]	○	◐	◐	○	○
Wilson [2005]	○	◐	○	○	○
<i>Handheld Systems</i>					
Beardsley et al. [2005]	○	●	○	○	○
Cao and Balakrishnan [2006]	○	●	◐	○	○
Cao et al. [2007]	○	●	◐	○	○
Raskar et al. [2006]	○	●	◐	○	○
Huber et al. [2012]	○	●	◐	○	○
<i>Wearable Systems</i>					
Mann [2000]	○	●	○	○	○
Harrison et al. [2011]	●	●	○	○	○
Mistry et al. [2009]	○	●	○	○	◐
<i>Document Management</i>					
Lamming et al. [2000]	◐	●	○	○	●
<i>AR-Based Systems</i>					
Oda et al. [2015]	○	◐	◐	○	◐
Gauglitz et al. [2014a]	○	◐	●	○	◐
Cidota et al. [2016]	○	◐	○	○	◐
Kuzuoka [1992]	○	○	○	○	◐
Ens et al. [2015]	○	●	●	◐	○

Table 7.1: Properties of selected systems for mobile work. ● indicates that a requirement is fulfilled, ◐ indicates partial fulfillment, and ○ that a requirement is not fulfilled.

R5 Information exchange Knowledge work processes often require communication with other people (e.g., through the exchange of documents). Sharing support should hence be provided by the systems.

7.3 Iterative Design Approach

The goal of this work was to explore the possibility of enhancing mobile working practices by providing an augmented-reality-based means of accessing the stationary physical workspace. To do so, we followed a user-centered iterative design approach. We started with an initial focus group in order to analyze the users' requirements that serve as a basis to inform the design of PiraTop. Based on the results, we developed an integrated interaction and visualization concept that we implemented prototypically. We then evaluated our prototype in a comprehensive final user study to assess the concept and inform the design of future systems. More concretely, we wanted to answer the following questions, which guide our work and also serve as a structure for the rest of this chapter:

- What kind of objects does the user want to access, and which actions does he want to conduct with them?
- How to visualize the physical environment and spatial layout to adapt to the remote environment?
- How to interact with the virtual representation of more complex objects like stacks?
- How does PiraTop perform?

The next section provides a first general overview of the architecture of PiraTop and presents the initial focus group that serves as a basis for the concepts introduced in the following sections.

7.4 PiraTop Overview

PiraTop aims to provide seamless integration between stationary augmented workspaces and mobile (knowledge) work. The PiraTop ecosystem consists of three main components, namely, the central data store and the user interfaces

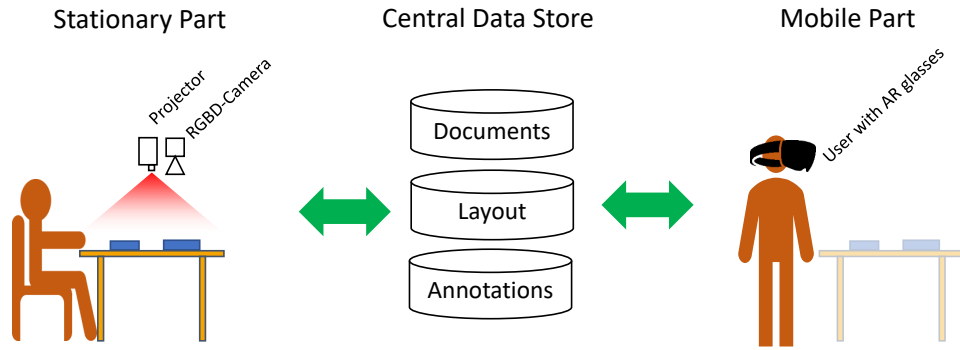


Figure 7.2: An overview of the PiraTop system. The stationary part is based on a hybrid tabletop environment, providing the data for the mobile augmented reality part.

for the physical workspace (stationary part) as well as the remote-access location (mobile part). An overview of the general structure of PiraTop is given in figure 7.2.

Central Data Store The central data store keeps the model of the workspace (i.e., a digital copy of all documents, their positions on the desk and digital annotations made to them). It is used to establish synchronization between the mobile part and the stationary part of PiraTop.

Stationary Part The stationary part, with its hybrid workspace, is the main anchor within the PiraTop ecosystem. Here, the workspace is physically present, and the user can carry out the full range of knowledge-work tasks in a convenient environment. An abstract setup for such a workspace is shown in figure 7.3. It usually consists of a set of integrated physical and digital documents that are piled (1,2), spread out (6) or in document trays (4), general

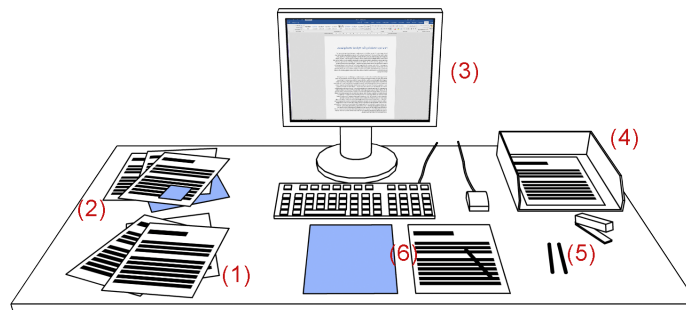


Figure 7.3: A hybrid workspace with purely physical (1) and hybrid (2) stacks, a desktop computer (3), a document tray (4), office tools (5), and two documents the user is currently working on (6). Blue denotes digital content.

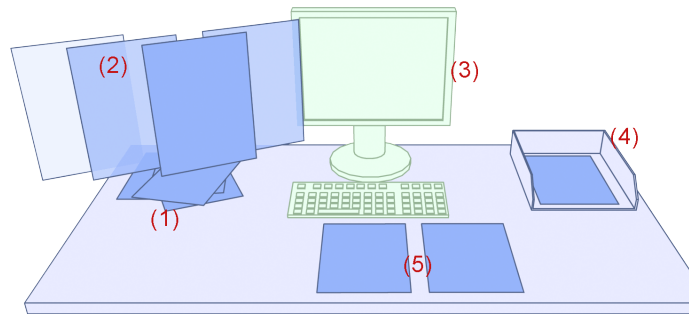


Figure 7.4: A mobile workspace with PiraTop with classic stacks (1), a spatially unfolded stack to facilitate browsing (2), large physical objects that serve as landmarks (3), a document tray (4), and single documents (5). Blue denotes interactive objects shown via AR, while green objects are passive proxies of physical objects shown via VR (e.g., to serve as landmarks).

office tools like pens or a stapler (5), and a conventional (desktop) computer (3). In order to provide the integration of digital and physical documents, hybrid tabletop systems are used, in our case the previously described StackTop system (c.f. chapter 4). The desktop computer is usually used for focused and deep engagement (e.g., writing) with a document, while the table is used to structure information and casually read or annotate documents. The digital model of the desk (i.e., which documents are present and their locations on the desk surface) built by the hybrid tabletop system is directly used by PiraTop to provide mobility for this workspace. Furthermore, augmented workplaces readily provide display facilities to visualize, for instance, digital documents or added digital annotations on the desk surface as well as on the surfaces of physical documents.

Mobile Part The mobile part allows the user, based on the workspace model, to access the workspace from anywhere he or she wants. To do so, PiraTop uses augmented reality to provide flexible, mobile, ad hoc way of access that provides a spatially correct view of the workspace so that the users can rely on their spatial memory and intuitively interact with the virtual workspace (see figure 7.4). The stack (1), documents (4), and tray (4) are presented the same way as on the physical desk (c.f. figure 7.3). While PiraTop aims at a natural representation, the virtual version of the workspace differs in a few aspects from its physical counterpart. Because PiraTop is aimed primarily at document remote access for information retrieval, it does not provide means of deep engagement (e.g., document writing or complex editing facilities). For this reason, the desktop computer, for instance, is visualized since it provides a prominent spatial landmark on most desks (3). Landmarks have been shown

to be an important factor for spatial memory, also in digital interfaces [Uddin et al., 2017]. However, it is not functional. Instead, using conventional mobile devices (e.g., tablets or notebooks) is supported for these applications. On the other side, PiraTop leverages the possibilities of AR to facilitate access, e.g., through more interactive visualizations of stacks (2).

Since the stationary part with tracking and projection on physical objects has been studied extensively in the previous chapters, we focus primarily on the mobile part with its visualization and interaction in the following.

7.4.1 Initial Focus Group

We conducted an initial focus-group session with two main goals: (1) to gain an understanding of interactions people would like to carry out while working remotely and (2) to refine and extend the initial requirements presented in section 7.2 in order to guide the design of PiraTop. The session had five participants from our department (three of them were employees, two were students) and lasted about two hours. All participants had at least some HCI-background and worked from home either regularly or at least occasionally.

During the session, four main areas of interaction were identified, which we will describe in the following.

Locating and Accessing Documents

This was seen as the most important category since it is the entry point for any further work to retrieve the necessary material or information. Also, it was found to be a frequent task to just quickly look up some information that is needed for an ongoing discussion without further working with it. As a result, PiraTop should provide a visualization of the work environment with its documents and spatial layout to facilitate finding information (R1). Further, it should also provide a way to conveniently work with structures present on working desks like single-page documents and stacks, which allows convenient browsing of the information as well as retrieving documents for further inspection (R2). Several possibilities were discussed; noteworthy were keeping the stack as naturally as possible as well as a coverflow-style way of browsing. Also, the participants wanted a way to take a closer look at documents easily (e.g., for retrieving a small piece of information like a figure) (R3) as well as a way to transfer the document to a notebook or tablet for

further inspection (R4).

Relocating Documents

One of the fundamental initial ideas of PiraTop is to leverage the spatial layout. Hence, what kind of interactions with respect to the layout are desirable was discussed. Generally, it was consented that one would ideally be able to move documents around just like one would do on a physical desk (R5). This would allow moving a document to one's focus area so that it is directly seen when returning to the stationary workplace or move it on a personal to-do pile or similar. While this is easy for digital documents, which can be directly moved by altering their locations in the virtual model of the workspace, there was no consent yet on how to deal with physical documents. The reason is that physical documents cannot be physically moved through interaction with an AR-based virtual proxy representation.

Working with Documents

Working with documents was another area of interest. Most prominent was the ability to add annotations to documents, e.g., highlighting areas or freehand drawing onto documents (R6). The ability to add new empty pages to later draw on them was also discussed; they could serve as virtual Post-its. Given the current state of AR technology and the aforementioned idea of transferring documents to a notebook, it was agreed that (if present) more intensive work with the document would occur there. In the other direction, a snapshot functionality was suggested, which would allow capturing a document, presentation slide, or other objects as an image that is then added to the PiraTop environment as a virtual document. Thereby, physical meeting notes could be directly digitalized and placed on the stationary workplace for further use (R7).

Collaboration with Others

Occasionally, participants wanted to share a document with other people (R8), e.g., in response to a phone call or e-mail. Being able to simply view another person's desk digitally was suggested. However, some participants were concerned about privacy if all documents are accessible. Also, it was noted that possibly, other people are unaware of a person's desk layout and would

perhaps put everything on the user's keyboard, which was, in turn, considered impolite.

Summary of Requirements

In summary, the requirements identified are the following:

- Req1 A visualization of the work environment with its documents and spatial layout to facilitate finding information.
- Req2 Convenient strategies for browsing and retrieving documents need to be provided to work with structures present on working desks, like single-page documents and stacks.
- Req3 A way to closely look into individual documents.
- Req4 Transfer of documents to a notebook or tablet to work on them.
- Req5 Interaction (e.g., moving) with documents should be similar to the behavior in the physical world.
- Req6 Annotation of documents, e.g., through highlighting areas, or freehand drawing onto documents.
- Req7 A snapshot functionality that allows capturing objects as an image.
- Req8 Sharing of documents with other people.

To support the identified areas and fulfill the requirements, PiraTop provides a set of interaction and visualization techniques that we will describe in detail in the following sections.

7.5 The Stationary Workspace

As mentioned initially, we based PiraTop on an existing augmented workspace system. The concept for the stationary setting is based on the StackTop system described in chapter 4. In order to incorporate the functions provided by PiraTop, the StackTop concept is expanded with the collaboration and remote functions described in the following, and summarized in table 7.2.

7.5.1 Mobile and Collaborative Functions

To support mobile work as well as collaboration with colleagues, two additional techniques (an in- and outbox system as well as a replay log) are provided that work for both hybrid stacks and individual hybrid documents. They are described in the following.

In- and Outbox

Sharing or handing documents with/to other persons is a frequent practice in knowledge work. As a result, this should be supported in PiraTop as well for both digital and physical documents, as discussed in our initial focus group (Req8).

A possible solution would have been to use AR to access another person’s desk just like one would do with their own desk using PiraTop. This would closely resemble real-world behavior where one can go to someone else’s office and place a document on their table. However, there are a few caveats to this solution: First, an informal survey among the staff of our department revealed that people find it often disturbing if someone just throws a document “somewhere” on the desk since it breaks their organization scheme. Second, having digital access to desks might cause privacy issues if everything is accessible or requires potentially complex access management.

Instead, we used a document-tray metaphor that uses the idea of different trays for different purposes – in this case, an in- and outbox. Having the physical or virtual boxes solved the two aforementioned issues since every user can place his inbox where his desk layout affords and a dedicated inbox provides direct visibility and awareness of new incoming documents. Also, sharing documents becomes an explicit action by placing documents in the outbox. Further, there are no privacy issues since the actual desk content is not accessible.

In the physical world, if someone passes on a document to someone else (e.g., handing it over personally or sending it via corporate mail), two possibilities exist: In case the original was handed over, it becomes inaccessible for the person handing it over. If the original owner wants to keep a version of the document, he has to photocopy it, resulting in two independent versions, of which one is passed along. PiraTop provides two different semantics in the context of the in- and outbox system, namely, *share* and *copy*. The *copy*

semantics are similar to scanning a document and then sending an email: the sender places a digital or physical document in the outbox, and the receiver gets a digital copy. The *share* semantics leverage the unique possibilities of digital systems by actually sharing a document so that, for instance, annotations made on one copy are transferred to the other users working on the same document as well. In both cases, after placing a document in the outbox, PiraTop asks with whom one wants to share the document (our prototype allows the user to select a receiver from a list of users) and whether the document shall be *shared* or *copied*.

Replay Log

Using PiraTop, users can move documents on their desk in AR (c.f. R5). While for digital documents, this is no problem since they can be relocated by the system, physical documents that were moved digitally remain at their physical locations. Although there has been research on automatically moving paper around [Ogata and Fukumoto, 2015; Probst et al., 2013], it is currently not a practical solution for general workspace use.

There are several possibilities to overcome this problem: The first option would be to prevent users from moving physical or even all documents when using the AR interface. While this would immediately solve the problem, it would either disable any spatial reorganization (which has been identified as important in the focus group) or lead to an inconsistent interface since some of the documents could be moved while others could not.

The second option is to generate a digital copy when moving a physical document. Hence, the document would be shown at the target location but, at the same time, still remain at the original place. Thereby, it would be possible to move around both types of object, but again, the physical object would remain at its original location, and the generation of copies is contradictory to the metaphor. However, this could be generally helpful for both document types if one just wants to remind oneself of a document by placing the copy somewhere within one's view without the intent to actually relocate it.

The third possibility is a kind of visual replay log where the system lists the move operations, and the user can carry them out manually when returning to his desk. While it requires the user to repeat the task of moving documents, this solution provides the highest flexibility. Later, if, for instance, e-Ink paper is more widespread, documents could as well relocate themselves.

Name	Purpose	Description
Outbox	Transferring documents to other users	By placing a document in the outbox, it can be shared with other users, either by sending a disconnected copy or by sharing it, which results in a synchronization of annotations.
Inbox	Receiving documents from other users	Documents received from other users are shown in the inbox. Thereby, they are received at a known well-defined location.
Replay Log	Applying changes to physical documents	Since physical objects cannot be moved by software, the replay log allows the user to manually apply the changes made to the location of physical documents by providing a list of changes.

Table 7.2: Summary of the interaction techniques included in the stationary PiraTop interface.

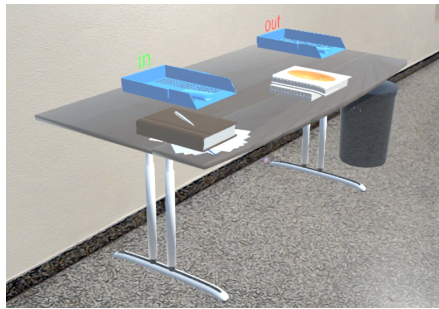
Trading off the properties of the different solutions, PiraTop uses the third variant because of its flexibility and consistency compared to the other two solutions.

7.6 The Mobile Part

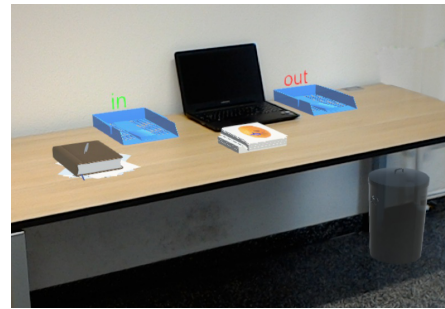
In this section, we focus on the visualization and interaction techniques of the mobile part of PiraTop. The provided techniques are described in the following. Table 7.3 summarizes the visualization techniques for stacks. The general interaction techniques are summarized in table 7.4.

7.6.1 Visualizing the Desk

The entry point for mobile working with PiraTop is an interactive virtual representation of the stationary workspace. As established in the initial focus group, a visualization that preserves the spatial layout and thereby facilitates information access is desirable (Req1). To be able to reproduce the spatial layout as naturally as possible, the mobile part was implemented using augmented reality. Thereby, it is possible to present the user a 3D replica of his workspace facilitating the use of spatial and potentially motor memory



(a) A workspace as complete AR visualization



(b) The workspace integrated with a physical table (table and laptop are real objects)

Figure 7.5: Augmented reality visualization of a physical desk with an in- and outbox, a trash can and documents.

when interacting with the workspace.

Integration with Existing Furniture

To allow for a convenient operation, PiraTop provides two modes for the mobile part: The first one, named *integrated mode*, can be used when working at another (empty) desk (e.g., in a shared office space). In this case, PiraTop augments the documents onto the existing desk surface (see figure 7.5b) such that they look like physically present documents. Integrating the remote layout with a physical layout present on the local physical desk is deliberately not supported, since it would interfere with the concept of conveying the original spatial layout to the user. Thus, only completely empty desks or desks with a sufficiently large contiguous empty area can be used with the integrated mode.

The second mode is the *green-field mode*, in which PiraTop provides a virtual desk as well, e.g., when using PiraTop on the go in a train or outdoors, where no physical desk surface is available. Another use case for the green-field mode is placing the virtual remote workspace right next to another physical workspace. For instance, in home-office settings, one could have the remote office desk next to the private desk (see figure 7.5a). The mobile part is described in the section after next.

Placement and Adaption

To place the virtual desk within the environment, PiraTop provides an automatic placement that determines a suitably large free area and moves the table there upon first use. The users can customize the position by moving the table

around. The location is then registered within the spatial model of the AR glasses so that the table remains static, even across multiple uses. Further, the user can decide to place the table on their own completely.

In some cases, available space is limited – e.g., in a train – so there is likely no available free space that is large enough for a desk. As a result, PiraTop needs to adapt the desk to fit the new location. For this, two basic possibilities exist: either keep the desk as it is and scale and rotate it to fit, or adapt the layout to the available space optimally. However, adapting the layout to available space (e.g., by changing the shape of the desk or compressing the layout on the desk so that a smaller table suffices) would break the usefulness of spatial memory, which is one of the core design ideas of PiraTop. To avoid this, PiraTop currently only supports moving, scaling, and rotating as options. In situations where not much space is available, it is likely that users also do not want to use extensive midair gesturing for interaction (e.g., when sitting on a train). In such scenarios, physical input devices (for instance, the HoloLens clicker) can be used to provide interaction capabilities with objects visualized within the personal space of other people without disturbing them.

Visualizing Objects

Given the desk visualization, the objects on the desk need to be visualized as well. PiraTop shows not only documents but also all other objects it is aware of since they can be important landmarks on a desk for structuring information. Further, they can carry information on their own, e.g., through their positions. However, if necessary, one could use these objects to reduce clutter by removing them in smaller scales if the user deems them unnecessary (e.g., the keyboard). To do so, the user would have to define which objects or which categories of objects are not needed as this is likely a personal preference.

Single documents can always be shown directly as paper-like objects. However, for stacks or folders, PiraTop uses a generic object representing a stack or pile that provides awareness of the size of the stack by being sized accordingly. Thereby, a visualized stack object does not actually consist of its individual documents. The reason is that working at a document-object level would be very cumbersome if one wanted to access a document deep within the stack or conveniently browse the virtual stack. To do so, PiraTop provides special visualizations described in the next section.

7.6.2 Browsing Stacks

Stacking and piling are important means of workspace organization [Malone, 1983]. They serve grouping purposes and allow the user to manage, for instance, different threads of work or different states of documents. As such, they play a key role for searching and accessing documents (Req2). In order to browse a stack, the stack needs to be visualized in an easily accessible manner as interacting with a stack on a single-page level like in the physical world would be cumbersome in AR environments.

An overarching goal of PiraTop is to facilitate the use of spatial orientation and memory by providing visualizations that preserve the spatial layout. This holds not only for the desk layout itself but also for the layout of a stack. This means that the position of a document within a stack should be correctly visualized as well. This allows, on the one hand, to have organization schemes that use a new-to-old order by putting new documents on top of an existing stack. On the other hand, more explicit schemes like alphabetical ordering are supported as well.

Since, to the best of our knowledge, no reference exists in the literature about suitable visualizations, we developed a set of four different visualizations based on known concepts from non-AR user interfaces (e.g., navigation of content in WIMP interfaces) that we will describe in the following sections and that are summarized in table 7.3. We later evaluated the visualizations in a user study. All four visualizations provide drag-and-drop reordering of documents within the stack as well as moving single documents from the stack to the table surface.

Tower

The first visualization, named *tower*, is close to a real stack in the way it visualizes the documents and their pages. Upon a tap-and-hold gesture and upward movement of the hand, the tower lifts a document, rotates it toward the user, then rotates it back to its horizontal orientation, and afterward moves it to a “second” tower above the initial one. Thereby, it creates the feeling of scrolling through the stack in a vertical direction. An advantage of this visualization is that the position within the stack is always clear to the user (see figure 7.6a) since both parts of the tower indicate the relative position. To avoid the tower growing too high, the “thickness” of the documents is scaled so that it remains within the user’s field of view. The tower visualization is

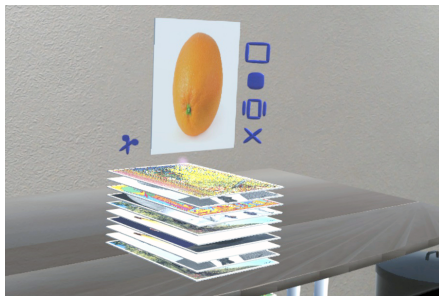
the only one of the four that retains the stack character. The other three visualizations use different arrangements in order to simplify interaction and visualization.

Coverflow

The *coverflow* visualization mimics a classic coverflow well-known from 2D GUI interfaces. It slides the documents horizontally in front of the user and, unlike the tower, where only one document can be viewed at the same time, provides some context by showing neighboring documents as well. Navigation is done using a tap-and-hold plus moving gesture, as it is the case for the tower. The coverflow visualization is depicted in figure 7.6b.

Carousel

The coverflow visualization is conceptually a more two-dimensional visualization. In order to leverage the possibilities of 3D space that AR provides, we implemented the *carousel* (see figure 7.6c). The documents are placed on a large cylinder in this visualization style. To navigate through the documents, the



(a) The tower visualization



(b) The coverflow visualization



(c) The carousel visualization



(d) The grid

Figure 7.6: Overview of the visualization techniques developed.

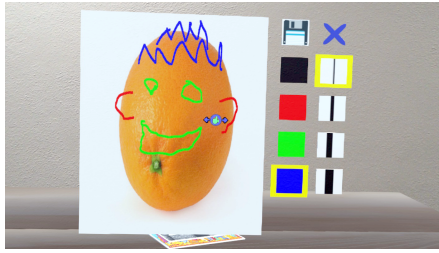
Name	Purpose	Description
Tower	Displaying and accessing a stack	The tower provides a stack-like visualization of a stack where the documents are visualized on top of one another. The user can navigate the stack via a dragging gesture.
Coverflow	Displaying and accessing a stack	The coverflow provides a horizontally aligned visualization of the stack, known from 2D GUI interfaces. The user can navigate the coverflow via a dragging gesture.
Carousel	Displaying and accessing a stack	The carousel visualizes the content of the stack on a cylinder that can be rotated with a drag-and-hold gesture. The cylinder can be rotated infinitely. Thus, the beginning is marked with a red bar.
Grid	Displaying and accessing a stack	The grid displays all documents within the stack at once in a static two-dimensional grid.

Table 7.3: Summary of the stack visualization techniques included in the mobile PiraTop interface.

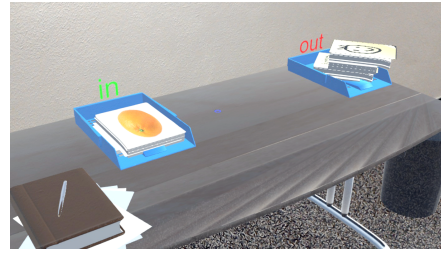
cylinder can be rotated by the user using a hand gesture, similar to coverflow, in order to explore the contents. Alternatively, the user could walk around the cylinder to explore the content since all documents are always in a fully visible state, unlike in the coverflow case, where the outer documents overlap. A red line marks the beginning (or end) of the stack since the carousel can be rotated infinitely.

Grid

As a baseline for the evaluation and because it is the most basic visualization, we further implemented a *grid* visualization that lines up the content of a stack in an $n \times n$ grid. While this visualization style possibly does not scale very well to larger amounts of documents, it has the advantage of providing a quick glance at the whole stack and facilitates drag-and-drop reorganization of the stack. See figure 7.6d for an example.



(a) The annotation mode allows drawing free-hand annotations



(b) The in/outbox system enables document exchange with colleagues

Figure 7.7: Overview of the document-level features of the mobile interface of PiraTop.

7.6.3 Detail View and Annotation of Documents

To support larger numbers of documents, the single pages are only shown at a comparably small scale that allows recognizing documents but is too small to actually read content using current AR glasses. Therefore, we implemented a detail view (Req3) that shows a larger version of a selected document upon tapping a page (see figure 7.7a). The detail view also serves as a basis for annotations.

Annotation of documents is a frequent task, e.g., to remind oneself to look at specific parts or highlight information for other persons (Req6). PiraTop provides annotation facilities as well in the form of freehand drawing on documents in AR. To do so, a set of colors and pen widths are provided. (see figure 7.7a). Drawing is done by a tap-and-hold gesture; in our HoloLens-based implementation, the pen follows the hand of the user. Depending on the technology used for implementation, other input devices (e.g., controllers) could be possible. To allow the transfer of annotations to the stationary workspace, annotations are stored separately from the document so that they can be transparently projected onto the physical counterparts. To create new notes, e.g. Post-its, the user can spawn an empty sheet that can be freely placed and annotated.

7.6.4 Sharing Documents

Sharing documents has also been identified in our initial focus group as an important action (Req8). We decided to implement this feature using an in- and outbox system just like in the stationary part of PiraTop. To share a document or set of documents, the user simply moves the document or stack into the outbox. In both cases, the document stays on the desk of the sending

user. We did not implement a move-semantic since it would be impossible for PiraTop to remove a physical document from the user's desk. This could, however, be easily added by using the replay log mentioned in the previous section to remind the user of bringing the physical copy to the recipient when returning to his stationary workplace. The AR visualization of the box system can be seen in figure 7.7b.

7.6.5 Transfer of Content

Generating new content is an essential part of knowledge work. It is hence important to be able to not only fetch documents from the remote workspace but also add new documents. In order to support both ways of transfer, PiraTop allows dropping a virtual document onto a physical laptop to transfer the document to that device for further editing, projection, etc. (Req4). In the other direction, it is possible to look at a physical document (e.g., meeting notes on a flip chart), capture an image of it, and then place it within the remote workspace (Req7).

7.7 Implementation

To evaluate the PiraTop concept, we prototypically implemented the AR part of PiraTop using a Microsoft HoloLens. The HoloLens was chosen because it provides a stand-alone augmented reality headset that is capable of running the required software. Using the environment model generated by the HoloLens, it is possible to register objects at fixed positions in the environment, which is beneficial for PiraTop as it allows placing the virtual desk at the desired location. Also, it already provides gesture recognition, although limited, as well as a physical clicker as input devices.

The PiraTop application is written in C# using Unity and runs entirely on the HoloLens, thereby eliminating the need for a cable to connect to a desktop computer. The application communicates with a web server over the Wi-Fi connection of the HoloLens to fetch the desk layout, documents, and annotation information that is stored in an XML-based format.

While working well for tidily aligned workspaces with suitable documents for tracking as it can be done using the PeriTop approach used in StackTop, automatically tracking an arbitrary physical desk in real-time is a complex and currently error-prone task. To provide a stable basis for the evaluation

Name	Purpose	Description
Desk Relocation	Relocating the desk visualization	The user can customize the automatic placement of the virtual desk by activating relocation mode. By doing so, the user can drag the desk around within the environment.
Reordering	Moving documents within a stack	All four visualizations support drag-and-drop reordering of documents within the stack. Also, it is possible to add or remove documents from a stack by dragging them from the table into the stack or vice-versa.
Detail View	Inspecting a document	The detail view shows a large version of a single document to allow the reading of content.
Annotation Mode	Annotating a document	In the detail view, the user can annotate documents by means of freehand annotations.
Sharing	Sharing documents with other users	Just like the in- and outbox in the physical workspace, the mobile part provides the same concept of in- and outboxes. See table 7.2.
Transfer of Content	Working with documents	It is possible to drag a document from the AR visualization onto a laptop, which leads to the document being transferred there for working on it. In the other direction, the user can capture an image of a physical object and use it as a document in the AR workspace.

Table 7.4: Summary of the interaction techniques included in the mobile PiraTop interface.

of PiraTop, we use a fixed manually created desk model for our evaluations. Thereby, all participants had the exact same state during the study, and problems like tracking/detection errors or communication failures could be avoided since the application could completely run on the HoloLens without external dependencies. However, it would be easy to generate the required XML files automatically if sophisticated document detection and tracking algorithms would become available.

7.8 Evaluation of the Visualization Styles

For the mobile augmented-reality-based part, we developed the previously described four different visualization techniques. Each visualization has different properties, namely, the size and number of documents shown, as well as the way the user interacts with them. Since one of the prominently named tasks in the initial focus group was retrieving documents, PiraTop should support

this as fast and conveniently as possible. As a result, we were interested to see how our visualizations perform regarding performance and cognitive load.

In order to compare the four visualization styles, we conducted a controlled experiment with our prototype. Our main measure of performance was time since one usually wants to retrieve a desired set of documents as fast as possible.

<i>Method:</i>	Quantitative
<i>Participants:</i>	19
<i>Duration:</i>	ca. 60 min
<i>Data Gathering:</i>	Logging, questionnaire, semistructured interview
<i>Variables</i>	
<i>Independent:</i>	Visualization, number of documents
<i>Dependent:</i>	Time, errors made

Table 7.5: Study design overview.

7.8.1 Experimental Conditions

The study consisted of four visualization conditions, namely, the tower, coverflow, carousel, and grid. Further, we varied the number of documents for each visualization to three different stack sizes (5, 10, 15) for each visualization since we assumed influence of the size of a stack on the performance of the visualizations as well.

7.8.2 Task and Experimental Design

To assess the performance of the visualizations, we used a counting task for our experiment. The users had to count the number of specific target documents within a larger set of documents. We used a corpus consisting of pictures of people, buildings, and fruits from which to draw the targets and distractors. The number of target objects was randomly chosen between 0 and 10 for each trial. The stacks were then filled with other objects to the number corresponding to the condition. There was a total of four stacks on the table in each condition.

Each condition (visualization with each stack size) was repeated three times, and the measurements were then averaged across the three repetitions. This leads to a total of $4 \text{ visualizations} \times 3 \text{ stack sizes} \times 3 \text{ repetitions} = 36$ recordings per participant. To account for learning effects, the order of the conditions was balanced using the Latin square approach.

After each visualization, the participants had to fill in a NASA TLX questionnaire to assess the perceived workload.

The completion time and errors (i.e., the deviation of the number the participant counted from the real number of targets) was automatically recorded. To do so, the participants hit a start button placed next to the desk in AR when they were ready to start, and then had a set of number buttons from 0 to 10 available to specify the answer once they felt they were done. The completion time recorded was then the time between the two button presses.

Before each session, the participants had some time to freely play around with the systems and familiarize themselves with the usage of the visualizations and the HoloLens in general. The actual task was started once they felt comfortable using the system.

7.8.3 Participants

The study had 19 participants, aged between 18 and 34 ($\mu = 25.3$, $\sigma = 4.54$), of which 4 were female. Of the 19 participants, 9 had non-computer-science backgrounds. Regarding prior experience with augmented reality, 4 had no prior experience at all, and 10 of them had prior experience with the Microsoft HoloLens. Hence, the results are more likely to be generalizable to a broader audience than computer scientists who are used to working with (new) technology. The participants were recruited from our department or

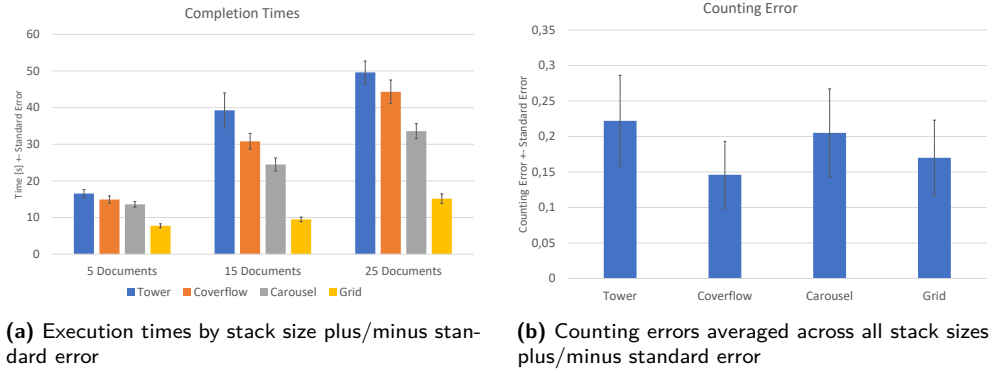


Figure 7.8: Results for time and error for the different visualizations.

students/friends of people working in our department. No compensation for participating in the study was provided.

7.8.4 Results and Discussion

We performed a two-way repeated-measures ANOVA to analyze the results of our experiment. Where necessary, Bonferroni correction was applied. In cases of violation of the assumption of sphericity, Greenhouse-Geisser correction was used.

Overall, the study led to a total of $19 \text{ participants} \times 4 \text{ visualizations} \times 3 \text{ stack sizes} \times 3 \text{ repetitions} = 684 \text{ recordings}$.

Completion Times

The visualization type had a significant influence on the time required to complete the task ($F_{1.798,32.370} = 60.116, p < 0.001$). The fastest visualization was the grid ($\mu = 10.801, SE = 0.614$), followed by the carousel ($\mu = 23.884, SE = 1.240$), the coverflow ($\mu = 30.006, SE = 1.904$), and, finally, the tower ($\mu = 35.131, SE = 2.646$). The results are shown in figure 7.8a. Pairwise comparison revealed that the differences between tower and coverflow are not significant ($p = 0.074$), while all others are ($p_{\text{coverflow};\text{carousel}} = 0.016$, all other $p < 0.005$).

As expected, we found a significant influence of the number of documents on the completion time as well ($F_{2,36} = 121.142, p < 0.001$), with the completion time increasing with the number of documents.

Further, there was a significant interaction for visualization \times number of

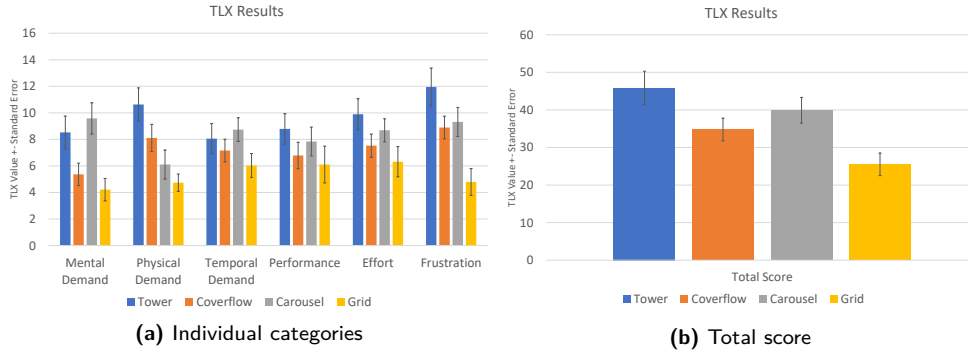


Figure 7.9: Raw TLX results by visualization type plus/minus standard error.

documents ($F_{3,159,56.858} = 15.628, p < 0.001$).

Errors

We found no significant influence of the visualization type on the errors made during counting ($F_{1,917,34.498} = 0.764, p = 0.468$), see figure 7.8b. There was also no significant interaction between visualization type and the number of documents ($F_{3,277,58.982} = 0.819$). When considering the absolute number of errors, the number of documents had a significant influence ($F_{1,410,25.377} = 5.134, p = 0.022$).

Upon looking at the relative errors (n_{err}/n_{docs}), we found no more significant influence of the number ($F_{1,151,20.713} = 0.57, p = 0.482$), while the influence of the visualization remains insignificant as well ($F_{1,613,29.039} = 1.284, p = 0.286$).

NASA TLX

For the NASA TLX, we found a significant influence of the visualization on mental demand ($F_{3,54} = 7.783, p < 0.001$), physical demand ($F_{3,54} = 9.202, p < 0.001$), temporal demand ($F_{3,54} = 2.824, p = 0.047$), effort ($F_{3,54} = 3.130, p = 0.033$), and frustration ($F_{3,54} = 10.429, p < 0.001$), but not on performance ($F_{3,54} = 1.271, p = 0.294$). The results are depicted in figure 7.9.

Taking a look at the total TLX score, pairwise comparison revealed a significant difference between the tower ($\mu = 45.842, SE = 4.462$) and grid ($\mu = 25.526, SE = 3.017$) with $p = 0.001$, as well as the carousel ($\mu = 39.895, SE = 3.422$) and grid with $p = 0.014$.

For the individual measures, significant differences were found for mental

demand between tower and grid ($p = 0.19$), coverflow and carousel ($p = 0.047$), as well as carousel and grid ($p = 0.047$). For the physical demand, the differences between tower and carousel as well as grid ($p = 0.012$ and $p = 0.003$, respectively) and coverflow and grid ($p = 0.022$) were significant. Temporal demand, performance, and effort exhibited no significant differences. Regarding frustration, the differences between grid and the three other visualizations (tower, coverflow, carousel) were significant ($p < 0.001$, $p = 0.03$ and $p = 0.003$).

Summary

Overall, since no significant differences exist in the number of errors users made across the visualizations and the average values were all rather low with values below 0.25, all visualizations can be concluded to be suitable to provide a sufficient overview on the content of a stack.

As expected, the grid was the fastest way for this task type since it gives an instant overview on the stack. However, it was noted by some participants that it will become confusing with more documents or when a detailed view of the document is required, e.g., to read text (P12: “*I’d only use it for short documents*”; P4: “*It’s good for photos*”). The reason is that 25 documents still fit conveniently in a 5×5 grid that is visible in the field of view of the HoloLens; however, for more documents, the grid might expand beyond what one can conveniently look at. While this is, to some extent, a hardware issue (e.g., the field of view of the HoloLens), there will be a point where some kind of “paging” must occur. This could be either actual discrete pages or a more continuous scrolling, which would, in turn, be addressed by applying some parts of the carousel or tower/coverflow visualization to the grid interaction-wise.

Interestingly, there is a significant time difference between the carousel and the tower and coverflow visualizations, while the speeds of the respective visualizations were aligned to be equivalent. Considering the observations during the evaluation and the user feedback, this is very likely caused by the fact that the tower and coverflow briefly “pause” if a document is in focus, while the carousel rotates continuously (e.g., P14 noted, “*It’s not fluent since it stops between changing documents*”). This leads to users releasing the dragging gesture and starting over, which requires additional time.

The aforementioned restarting of the gesture is reflected in the physical demand of the TLX that is lower for the carousel and grid than for the tower (and coverflow, although not significantly). Therefore, changing the animation

style such that it looks more smooth and does not foster “restarting” the gesture has to be considered. The brief stopping of the animation was a design decision made to facilitate stopping at a specific position for closer inspection of a document. Otherwise, since the document first rotates from a horizontal alignment to a vertical one and then back while moving from one part of the stack to the other, the document might stop in a rotated intermediate position. Based on the user feedback, a possible solution would be to remove the stopping and use a fluent animation and only align the current document after the user stops the scrolling gesture. Thereby, the notable stopping would be eliminated.

The study revealed that the coverflow and grid exhibit a similar mental demand, while the tower has a significantly higher mental demand. The high mental demand of the carousel results from the carousel not having a start marking or position indicator in the first version used in the study. This made it difficult to see where it starts over. This is in line with the participants’ feedback (P3: “*It would be much better if there was a start marking and position indicator*”; P7, P9, and P14 made similar statements). While there are a few significant differences in the individual scores, the total TLX score is only significant with respect to the grid.

In summary, the type of visualization desired by the users is likely to be task dependent: While the grid was the fastest, users noted that other visualizations are better suitable to get a closer look at individual documents and that the grid might become confusing with more documents. As a result, further studies that consider different task types instead of only a counting task are required. Therefore, the following study will look into visualization use for different task types.

7.9 Evaluation of the Mobile Part

<i>Method:</i>	Qualitative
<i>Participants:</i>	18
<i>Duration:</i>	ca. 60 min
<i>Data Gathering:</i>	Questionnaire, semistructured interview

Table 7.6: Study design overview

To gain overall insights into how PiraTop is received by users, we conducted a qualitative study using our prototype system. For this evaluation, we gave the users a set of tasks that involved the use of all system features. The tasks were selected to be similar

to situations that would be encountered in productive use. This is contrary

to the first evaluation, where the main goal was to compare the performance of the different visualizations. Participants were asked to “think aloud” in case they encountered problems, and we collected user feedback afterward in a semistructured interview. Finally, we used the AttrakDiff questionnaire to assess the overall perceived quality and attractiveness of the system.

7.9.1 Experimental Conditions

For this evaluation, there was only a single condition: the PiraTop system without any restrictions. We used the same document set as in the previous study, from which we generated four stacks suitable for the tasks. Two of the stacks consisted of 10, the two others of 15 documents. They were identical for all participants. Also, unlike in the previous study, we did not impose a specific visualization to be used. Users were free to choose the visualization they felt to be suitable for the task.

7.9.2 Tasks and Experimental Design

The study was designed along a set of specific tasks leading to the use of a wide range of functionality of the mobile interface of PiraTop.

The set of tasks for the study was as follows and had to be done in the presented order:

1. Out of the four stacks lying on the table, find the one that contains an orange and share it with another person via the outbox.
2. Use the (read-only) document received via the inbox to answer a set of questions regarding three documents within the full set. Further, highlight the answers using the annotation functionality.
3. In the (editable) document received via the inbox, reorder the exams by matriculation number in ascending order.
4. Use the file opener to load three additional sets of documents. Put the one with fruits in it into the trash.

The experiment was designed as a single-run study – i.e., there were no repetitions of the tasks. The users were again allowed to play around with all the visualization types, the document opener, trash can, and the in- and

outbox system freely, until they felt comfortable using it and were ready to start the actual tasks. While conducting the study, the participants were free to stand up and walk around or to use the clicker instead of hand gestures. There was no time limit or other constraints imposed.

7.9.3 Participants

The study had 18 participants, aged between 22 and 35 ($\mu = 26.4$, $\sigma = 3.65$), of which 5 were female. Of the 18 participants, 7 had non-computer-science backgrounds, 5 had no prior experience with augmented reality at all, and 12 of them had prior experience with the Microsoft HoloLens. 8 of the 18 participants had already participated in the visualization study. As in the previous study, we expect a broader range of feedback than we would get from computer-science-only participants or regular augmented reality users. The participants were recruited from our department or students/friends of people working in our department. No compensation for participating in the study was provided.

7.9.4 Results and Discussion

AttrakDiff Questionnaire

To get an understanding of how the participants perceived the augmented reality user interface of PiraTop in general, we had them fill in the AttrakDiff questionnaire [Hassenzahl et al., 2003] after completing all tasks.

In nearly all dimensions of the AttrakDiff questionnaire, the assessment by the participants was on the positive side on average, as depicted in figure 7.10a.

Concerning pragmatic quality, the system has most prominently been found to be clearly structured ($\mu = 1.667$, $SE = 0.377$), predictable ($\mu = 1.167$, $SE = 0.362$), and manageable ($\mu = 1$, $SE = 0.333$). While it was also found to be straightforward ($\mu = 0.556$, $SE = 0.370$) and simple ($\mu = 0.5$, $SE = 0.425$), on the downside, there was a slight feeling for it to be impractical ($\mu = -0.222$, $SE = 0.389$) and technical ($\mu = -0.5$, $SE = 0.336$).

For the identification part of hedonic quality, the participants found PiraTop to be professional ($\mu = 0.444$, $SE = 0.439$), stylish ($\mu = 0.833$, $SE = 0.362$), premium ($\mu = 0.5$, $SE = 0.403$), and integrating ($\mu = 0.778$, $SE = 0.3$). However, they felt it was isolating ($\mu = -0.167$, $SE = 0.425$) and separating

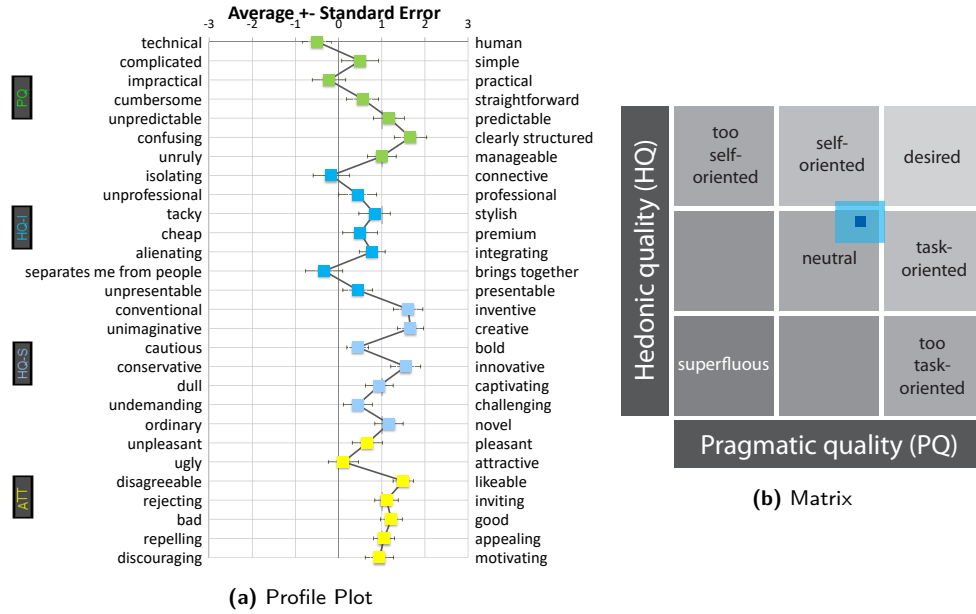


Figure 7.10: Results from the AttrakDiff questionnaire for PiraTop.

them from people ($\mu = -0.333$, $SE = 0.430$).

Stimulation in the sense of hedonic quality, on the other hand, proved to be completely positive, with high values for invention ($\mu = 1.611$, $SE = 0.344$), creativity ($\mu = 1.667$, $SE = 0.304$), and innovation ($\mu = 1.556$, $SE = 0.344$). The system has also been rated novel ($\mu = 1.167$, $SE = 0.326$) and captivating ($\mu = 0.944$, $SE = 0.319$) as well as slightly challenging ($\mu = 0.444$, $SE = 0.335$), and bold ($\mu = 0.444$, $SE = 0.251$).

In terms of attractiveness, the system is in between ugly and attractive ($\mu = 0.111$, $SE = 0.350$) but more pleasant than unpleasant ($\mu = 0.667$, $SE = 0.351$). On the other hand, it has been found to be likable ($\mu = 1.5$, $SE = 0.239$), inviting ($\mu = 1.111$, $SE = 0.271$), good ($\mu = 1.222$, $SE = 0.255$), appealing ($\mu = 1.056$, $SE = 0.242$), and motivating ($\mu = 0.944$, $SE = 0.328$).

This leads to a good overall assessment of PiraTop; on both the self-oriented and task-oriented dimensions, it has a positive tendency toward “desired”, while not yet reaching it completely (see figure 7.10b). The most negative attributes were that it is too technical, slightly impractical, isolating, and separating from people. Regarding ugliness/attractiveness, the rating was neutral. When taking the general feedback into account, some of the negative associations are likely a result from the prototype stage (especially the lack of attractiveness) of the implementation as well as the limitations of the interaction through the HoloLens, with basically only one gesture available

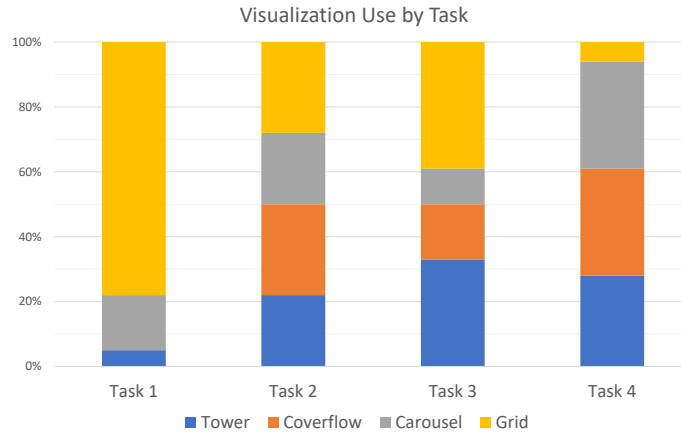


Figure 7.11: Visualization distribution by task.

for interaction (impracticality and PiraTop being too technical). However, the results are promising and show that having virtual augmented reality access to one’s workspace is generally a well-perceived concept.

Preferred Visualizations

Since the users were completely free in this evaluation to choose the visualization they preferred and to change it as desired, we were interested to see if there was a visualization preference for certain tasks and whether the users bothered at all to change to a visualization they found suitable for a task.

Indeed, we found that participants preferred different visualizations for different tasks, as shown in figure 7.11.

In the first task, 78% of the participants used the grid visualization, 17% the carousel, 5% the tower, and 0% the coverflow.

For the second task, only 28% used the grid, while the rest used one of the other three, leading to 28% for the coverflow, 22% for the tower, and another 22% for the carousel.

For the third task, 39% used the grid, 33% the tower, 17% the coverflow, and 11% the carousel.

In the last task, only 6% chose to use the grid, while there were 28% for the tower, 33% for the coverflow, and another 33% for the carousel.

In summary, we can say that although the first study revealed that the grid is the most performant visualization time-wise and the least demanding based

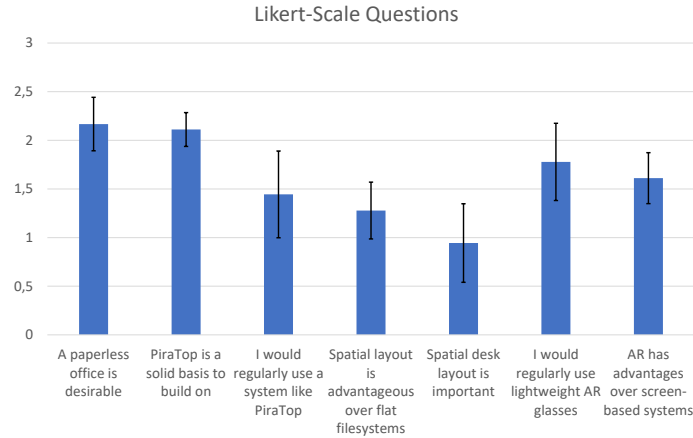


Figure 7.12: Results of the Likert-scale questionnaire; answer possibilities ranged from -3 to 3.

on the TLX, users also prefer to use other visualization styles depending on the task. If we take a look at the numbers in relation to the tasks, there is a strong focus on the use of the grid in tasks where an overview on the whole stack is required. When it comes to working with individual documents, the other visualizations are chosen as well. This might be due to individual documents becoming rather small in the grid for higher document counts, while for the other visualizations, the size remains the same independent of the number of documents.

Questionnaire Results

The questionnaire consisted of a set of Likert-scale questions (range -3 to 3 (totally disagree to totally agree); see figure 7.12).

All participants agreed that a paperless office or office with less paper is overall desirable for environmental reasons, easier data management and sharing, etc. ($\mu = 2.167$, $SE = 0.275$). Most participants liked the general idea of PiraTop and saw the prototype as a solid basis for further development ($\mu = 2.111$, $SE = 0.174$). They also would like to regularly use a space-aware paperless augmented reality system like the one in this study for work if it was more polished ($\mu = 1.444$, $SE = 0.446$) and said that spatial document placement could have advantages over flat file-system views ($\mu = 1.278$, $SE = 0.292$). However, when explicitly asked, the spatial layout of their desks was important for some of the participants, depending on whether they work with more paper-oriented or more digital-oriented methods ($\mu = 0.944$, $SE = 0.404$).

Since PiraTop requires the use of augmented reality headsets, we asked the participants whether they would be willing to use them in their daily life as well. Most of the participants would be willing to use augmented reality headsets in everyday life if they were as light as regular glasses ($\mu = 1.778$, $SE = 0.397$) and think that augmented reality can have advantages over screen-based systems ($\mu = 1.611$, $SE = 0.262$). Only a few commented negatively on the use of AR in everyday life, e.g., because of the socially awkward interactions with the glasses through midair gestures.

In summary, we can say that users generally like the idea of using AR-based systems, even on a regular basis if the devices evolve to be lightweight and comfortable to wear. The idea of having a remote access to stationary workplaces was well received as well, although there are different opinions on the helpfulness of spatial layout. This, however, is likely to be a result of the user's habits.

Qualitative User Feedback

The outcome of the previous sections is also reflected in the interviews. For the open questions, we asked the participants what they liked and did not like and whether they had any improvements.

Positive Aspects On the positive side, the participants said that they liked the idea in general and swiping through and editing documents in augmented reality were found to be “cool”. They found it generally positive to have various forms of visualizations to choose from depending on the current situation and found them visually appealing. The grid reminded them of a directory view where one can see everything at once and use it as a good entry overview. The carousel has been seen as another overview that is especially appealing. Being able to edit pages starting from each visualization was also well-received.

One participant saw the system as an all-in-one solution for viewing and sharing documents. The ability to interact with not only their own physical desk but also with other people has been liked, as well as that sharing was easy to use with only two steps. The sharing system provided by PiraTop has been seen as useful not only for large office scenarios but also for life on the go.

The paper stacks have been seen as appealing and easy to place and open. One participant noted that throwing around stacks in the room was good for reducing stress.

The participants have overall attributed the system as novel, interesting, innovative, futuristic high technology. To most participants, it was easy to understand and use; the design has been found to be plain and simple, with good dimensioning of objects and fluent interaction. One user deemed it ready to be used as soon as it is more polished, also by companies. The skeuomorphic design was liked; it felt familiar for one user. Like on a real desk, data could not be forgotten.

User Critique On the other side, there was also negative feedback concerning some aspects of the prototype. The horizontal next and previous pages of the tower were regarded as too high (i.e., consuming too much space), the carousel was seen as too fast by one user. These are, however, no issues with the general concept but rather a result of the prototypical implementation and its configuration. The speed of the carousel, for instance, can easily be adjusted. In a more sophisticated implementation, such parameters would be configurable. One user found that all visualizations except the grid were only nice to play around with, but “essentially useless” (P8).

Interaction sometimes seemed to be unnecessarily complex and was seen as “laggy” by a few participants. In the same direction, the interaction has been criticized as being unreliable and fiddly, requiring difficult gestures as well as head and hand coordination. This is, however, not only a direct problem of the PiraTop concept but also a result of the underlying hardware used for the implementation. The same holds for the complaint of one user who described the pages to be too blurry, small, and low resolution, making reading difficult. This is again a problem of the current technological state of augmented reality, which will likely be resolved by future generations of AR hardware. One participant noted that there is no added value in reading in augmented reality instead of transferring the document to a notebook or tablet.

Explicitly mentioning the HoloLens, some participants found the hardware too heavy, the field of view too small, the resolution too low, and the air tap too difficult. One participant said the idea of PiraTop was great, but HoloLens was the problem.

Suggested Improvements Among the suggestions for improvements, there was the wish for a more paper-like interface where the paper is lying on the table instead of flying or can be held in hand with an appropriate holding gesture. To improve the visualizations, a “one-dimensional grid” (i.e., lining

up all elements in a row) has been requested as well as a “multi-dimensional carousel”. One participant wanted to place multiple visualizations manually or intelligently automatically.

For the annotation function, participants wanted to use a pen rather than their bare hands and to be able to mark a whole word by tapping it. The annotation drawing lines should have semitransparency and should be easier to draw in a precise manner. A back button or eraser has been requested.

Participants also suggested to add an additional board to pin documents in case the desk was full or even no desk at all but freely flying documents that move along with oneself. One suggestion was also to make the whole area around the desk behave like a document shredder.

For interaction, people wished to have more sophisticated gestures. More feedback and even sound would have been welcomed.

Support for more use cases has been requested as well by one participant who wanted not only to access documents but also use Pira_{Top} as a virtual snack bar, where taking a virtual snack from the table would lead to an order of the snack in real life, like a pizza box connected to a pizza-delivery company.

Summary In summary, we can say that the users found Pira_{Top} to be an easy-to-use solution for remote access and sharing content with other users. The availability of various different visualizations for stacks was seen as positive since it allows the user to select the visualization most suitable for the current task. The general concept was well received by the users.

However, there were a few aspects the users criticised. Some of them are results of the current state of technology for head-worn AR glasses (e.g., the weight or the small field of view). These are likely to be resolved with future generations of such devices. The main points of critique that are related to the Pira_{Top} system itself were not regarding the fundamental concepts, but rather concerning specific aspects of the implementation (e.g., scrolling speed, size of the elements). Also, they can be easily solved in a next iteration of the implementation. This is in line with the general positive statements above and the results of the questionnaire.

Finally, the participants suggested to add a richer interaction vocabulary with more differentiated gestures, support for pens and erasers to ease creation of annotations and general improvements of the visualizations.

7.10 Design Implications

Based on the results of our evaluations, we derived a set of design implications for augmented-reality-based knowledge-work environments.

7.10.1 Spatial Layout Matters

The spatial-layout-preserving way of presenting the remote desk was generally acknowledged as useful, albeit subject to different personal preferences. Only a few participants who work mostly with digital documents and hence know their location on the computer very well but are not so aware of their desk layout found it not explicitly useful. A reason for this could be that people who work primarily with digital documents adapted their work practices to the limited organization strategies provided by current desktop operating systems (e.g., folders and file names). As a result, they might not see any advantage of spatial organization as they are not used to it. This, however, does not mean that it could not turn out to be beneficial for such users as well if they would adopt such strategies. For this to happen, it would likely be beneficial if spatial organization would be employed more prominently in desktop operating systems as well. For all other participants, it was the other way around. This is in line with previous research and likely depends on the way people organize their work [Kirsh, 1995; Malone, 1983]. As a general guideline, it is beneficial to provide the spatial information since it is helpful for people who rely on this in their real-life work environment while it is not seen as negative by people who do not find it strongly beneficial in the physical world.

7.10.2 Different Visualization Styles Should be Provided

As expected, the evaluation of our visualization styles showed that the grid is the fastest visualization style when it comes to finding objects. On the other side, the second evaluation showed that in real-world tasks, participants also used the other visualizations. Especially for tasks where a closer look at a single document is necessary, the grid was not as dominant as in the overview task. In these cases, the carousel, tower, or coverflow have the advantage that they always have the same size regardless of the number of documents within a stack. Hence, the individual documents are visualized larger than within the grid, which has to scale down the individual documents to still fit within the field of view with the increasing number of documents. As a result, it is

desirable to provide different visualizations, at least one that provides a general overview and one that allows for a closer inspection of individual documents.

7.10.3 A Rich Interaction Vocabulary is Desirable

The need for space-consuming and complex midair gestures has been found to be socially awkward [Rico and Brewster, 2009, 2010] and fatiguing [Hincapié-Ramos et al., 2014] in previous research. Hence, our initial concept for PiraTop relied only on the simple hand gestures provided by the HoloLens and, as an alternative, the HoloLens clicker. However, during the study, it turned out that participants would have preferred to additionally have a more natural way of interaction as well. They would have liked to use grasping or swiping gestures that resemble more what one would naturally do in the physical world. Especially when a real-world-based visualization is used, it is hence advisable to provide the interaction gestures known from the physical world in augmented reality. These gestures should be as close as possible to real-world behavior as well. To support scenarios where users do not want to use expansive gestures and to avoid conflicts between natural and space-reduced gestures, it might be sensible to either allow the user to manually switch to a reduced gesture set or detect the situation and interaction intent automatically and adjust the used gesture set on this basis.

7.10.4 Physical Proxies can Enrich Interaction

A few participants complained that it was unnatural to interact with “paper” objects that are not actually graspable. Therefore, it might be desirable to provide a haptic dimension to the interaction with the AR environment. To do so, several possibilities have been explored by other researchers, such as using vibro-tactile feedback [Park et al., 2011], ultra-sonic transmitters [Spelmezan et al., 2016], or even electric muscle stimulation [Kajimoto et al., 2004]. However, these solutions require external devices placed in the environment (ultrasonic) or worn by the user (vibro-tactile feedback and electric muscle stimulation). We therefore suggest based on the feedback of one participant the use of a kind of proxy paper object that would be used to bind virtual documents to. They would then be visualized on the surface of the physical paper object. Thereby, the interaction (e.g., moving or annotating) would feel more natural. At the same time, the proxy paper is generic so that a user would only have to carry the AR glasses and a few sheets of paper with him.

7.11 Summary and Conclusion

In order to address the challenges of mobile work, this chapter presented a remote access solution for the ubiquitous trend of knowledge workers to work not only at a fixed office but instead at various locations while on the move. A common issue when working on the go is the lack of access to information, which is especially crucial for knowledge workers. Using mobile devices, access to digital information is possible nearly anywhere today. For information stored on physical media, e.g., paper, due to its weight, the user can carry along only a limited amount of documents that he has to select upfront.

Considering the concepts for stationary hybrid knowledge work presented in this thesis, this becomes even more problematic. Using the presented concepts, the user is now able to seamlessly interweave and spatially arrange digital and physical documents on his stationary working desk. However, taking along such mixed sets of documents is difficult, since the physical part would need to be carried along in a briefcase while the digital part would need to be transported using a laptop or tablet. Thereby, any spatial arrangement is destroyed.

In order to overcome these limitations, we presented PiraTop, an integrated concept for augmented-reality-based mobile work. Based on an initial focus group, we derived concepts for visualization and interaction with remote workspaces that allow the user to leverage his or her spatial memory to foster productivity and ease access to information. The PiraTop concept enables the user to interact with his hybrid desktop environment from remote locations. Concepts to retrieve documents or interact with them in their known spatial layout are provided. We then conducted two studies to evaluate the different visualization styles developed as well as the augmented reality part of PiraTop as a whole. The evaluation proved that the general concept of providing a spatially correct desk visualization used in PiraTop is well received by the users. The main points of critique were related to the hardware used and parameterization of the visualization and interaction techniques (e.g., scroll speeds). These are, however, easy to solve in future iterations. Based on the participant's feedback, we then derived a set of design implications to guide the future development of augmented reality workspace systems.

Since PiraTop requires that the user has AR glasses for mobile use, a salient question is whether the use of AR in stationary settings could be beneficial as well. To that end, we propose a concept for the use of AR glasses in the context of a stationary PeriTop environment in the next chapter.

Part V

Conclusion and Outlook

OverTop – PeriTop Revisited

The past chapters presented approaches fostering a tighter integration of digital and physical documents as well as tangibles in hybrid tabletop environments. By means of top-projection on physical objects, documents, and tangibles, the display space broke the confinement of the glass screen and extended the interface into the third dimension. However, the content still mostly sticks to the screen or is projected onto the surface of rigid objects. This chapter revisits the PeriTop concept and introduces OverTop, a concept for providing a full three-dimensional interaction experience for both content and user interface. OverTop introduces three different layers, in which both content and interfaces can be presented: We coin these layers as *Below*, *On*, and *Above*. Using these three layers, OverTop extends the design space of tabletop interfaces to the area above traditional tabletop workplaces, opening up new interaction and visualization possibilities.

After an introductory section, we present the OverTop concept and the properties of the three layers in section 8.2. Section 8.3 demonstrates the practical applicability of the OverTop concept by presenting five interactions that become possible through OverTop. The chapter concludes with a summary (section 8.4).

8.1 Introduction

In the past chapters that form the main part of this thesis, we presented approaches to fostering a tighter integration of digital and physical documents as well as tangibles in hybrid tabletop environments. To do so, we presented the PeriTop and FlowPut concepts that leverage the surfaces of documents and tangibles as additional top-projected display and provided touch input in order to make the display interactive. Besides saving space on the primary display, the concepts presented broke the confinement of the glass screen and extended the interface into the third dimension. However, the content still sticks to a

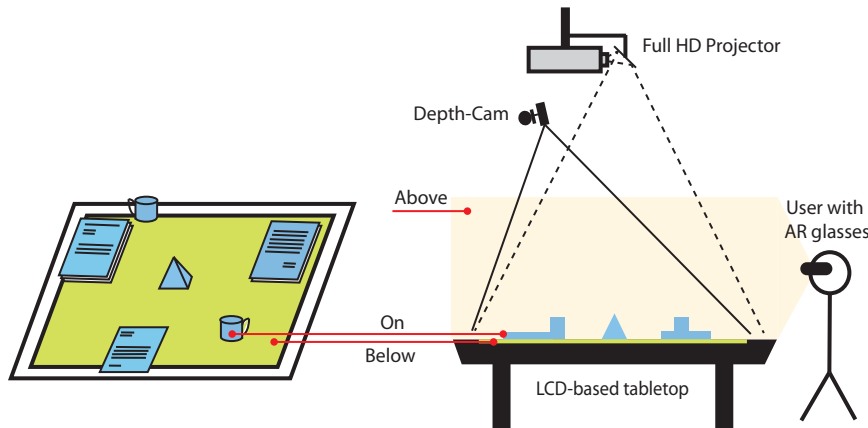


Figure 8.1: The three-layered interaction space of OverTop and its general hardware setup.

two-dimensional surface that is either the screen or the projection onto the surface of rigid objects. In this chapter, we revisit the PeriTop concept and introduce OverTop, a concept for providing a full three-dimensional interaction experience for both content and user interface that is based on the use of augmented reality.

With the advent of consumer-usable AR glasses, it is possible to break the confinement of flat surfaces and visualize dynamic digital content in 3D on and above the tabletop surface. Thereby, the possibilities of usually static physical tangibles are extended. Adding AR to the conceptional setup of PeriTop allows for both, richer visualization and interaction techniques as the third dimension can be used. For example, a common problem using conventional touch to move objects in hybrid tabletop settings is that the direct path is often obstructed, and one needs to find a way around the obstacles. In 3D space, the object could simply be lifted and moved over the obstacle as one would do with a physical sheet of paper. At the same time, projection and active displays are still beneficial because they exhibit different properties with respect to public/privateness and visual properties as AR.

8.2 The OverTop Concept

Conceptually, OverTop splits the interactive space into three layers, of which each has unique properties and roles. The overall layer structure and setup of OverTop is depicted in figure 8.1.

As mentioned in the introduction of this chapter, the past contributions of

this thesis focused on extending a conventional tabletop display (the *below layer* in the following) by adding the surfaces of objects as additional peripheral display space (the *on layer* in the following). While at a first glance, this might seem unnecessary, the contributions and the evaluations presented along them have proven the usefulness of the *on layer* that is provided by the PeriTop concept. The OverTop concept now goes an additional step further by introducing another layer, namely, the *above layer*. The above layer is the first layer that is actually capable of natively displaying 3D information by means of AR. Thereby, it opens up additional possibilities for information presentation as well as interaction that we will demonstrate in the next section.

In the following, we will give an overview of the different layers and their properties.

8.2.1 Below Layer

The *below layer* is basically the display embedded into the table surface and thus below any physical object placed on the table. While providing an intuitive tabletop display, traditionally, it is impossible for the user to move any object on this layer above any object placed on the surface (hence below layer). It serves as the primary layer for self-contained digital objects, like documents, which are well known from classic digital tabletops. The display surface is inherently flat; hence, it is limited to two-dimensional content. However, since it can be based off an active LCD, it provides high-resolution display capabilities. Also, it serves as an inherently public display and can, therefore, for example, be used to share content with conversation partners.

8.2.2 On Layer

The *on layer* is an additional content layer that resides on the surfaces of any physical object placed within the interactive space. It can be used to enrich the objects, e.g., through annotations or contextual information. While the projection surface has a three-dimensional structure, the content projected is still two-dimensional. Unlike the below layer, the on layer must cope with interference from the original texture of the objects. Also, given that projection is often used in this context, the resolution is lower, and the display is more prone to daylight interference than the active LCD. Furthermore, projection is affected by shadowing through body parts and tall objects on the table. Just like the table surface, the display space is public.

	Below	On	Above
Private/Public	public	public	both
Structure	2D	2.5D	3D
Resolution	4K(/8K)	1080p/4K	“full HD”*

Table 8.1: Summary of the layer properties. *) Non-standard resolutions, roughly equivalent to full HD.

8.2.3 Above Layer

The *above layer* is provided by an AR system and comprises the airspace above, and even around, the table. It is used to display three-dimensional information that can be freely placed. While the first two layers are inherently public, the above layer can be used to display both private and public information since a personal device performs the visualization. The transition among these three layers can be used to additionally enrich interaction, as described in the next section.

8.2.4 Layer Properties

Table 8.1 summarizes the properties of the different layers that we will discuss in the following.

Private/Public Interaction Besides the possible use as a single-user personal working desk (e.g., in the office), tabletops are also used in collaborative settings with multiple users. While privacy of information is not a predominant issue in the single-user setting, its importance increases in collaborative settings. As such, research has investigated how to personalize input and output on tabletops [Lissermann et al., 2014] but considering only the *below layer*. In our basic concept, just the *above layer* may be personal, while the other two layers are always public. Since the *above layer* can add content to the other two layers as well by overlaying, this does not limit the possibilities of OverTop to show private information. Yet, the use of inherently public display technologies or surfaces poses a set of advantages, especially in asymmetric scenarios. For instance, while a presenter during a presentation sees additional details, viewers may only see the content with some projected augmentation on important features. In this case, most of the users do not necessarily need AR glasses since the *below* and *on layers* are inherently public.

Display Structure The different layers have different structures that need

to be taken into account when choosing where to display information. The *below layer* is two-dimensional since it is a flat display. Leaving the flat surface, the *on layer* extends display space to structured surfaces that protrude from the table surface. While, as a result, the display surface now has a three-dimensional structure, the displayed content is still two dimensional. Therefore, we consider the *on layer* as 2.5-dimensional because of the added height. The *above layer* in contrast provides full three-dimensional display capabilities that allow displaying 3D content natively.

Resolution Contrary to the increasing degrees of freedom in display structure, current technology leads to a decreasing resolution of the layers from below to above. While the technology may improve in the future, currently and in the near future, one will have to consider the resolution of the layers. The tabletop itself provides a very high resolution for the *below layer* (current displays offer a 4K resolution, with 8K being around the corner). For the *on layer*, a full HD to 4K resolution is possible with projection technology that is common today. Lastly, the *above layer*, realized using AR glasses, has the lowest resolution with roughly full HD being common today. As a result, one would, for instance, preferentially display longer textual content on the *below layer* since it is 2D content that does benefit from a high resolution. On the other hand, a 3D CAD model would be displayed in the *above layer* since it is 3D content and can be viewed at a lower resolution.

8.2.5 Technical Implementation

While we currently use an active LCD tabletop for the *below layer*, a projector for the *on layer*, and a HoloLens for the *above layer*, the concept is generally technology agnostic as long as it is possible to implement the respective layer properties. For instance, if AR glasses were more widely spread, one could realize the scenario completely within AR. Yet, the current level of technology does not allow doing all this at the same level of sophistication. Therefore, we propose to use this combination to implement the OverTop concept.

8.3 Interactions Enabled by OverTop

In this section, we describe a set of interaction techniques leveraging the possibilities of OverTop.

8.3.1 Ad Hoc Data Visualization



Figure 8.2: Ad hoc data visualization for selected content.

When reading documents, there are cases in which the given representation of data is not practical for the current intention of the user (e.g., one might find a table of sales figures that is difficult to grasp if one just wants a quick glance at the development of sales). We, therefore, propose to provide an ad hoc way of selecting data from documents (digital or physical; cf. the DigitalDesk [Wellner, 1993]) and generating 3D visualizations thereof (e.g., diagrams

or simulations). The *below layer* is used as document display for digital documents, the *on layer* is used to augment highlights on physical documents to visualize the selection process, and the *above layer* is used for the actual display. Figure 8.2 shows an example of visualizing sales numbers from a selected part of a physical document.

8.3.2 Label Display

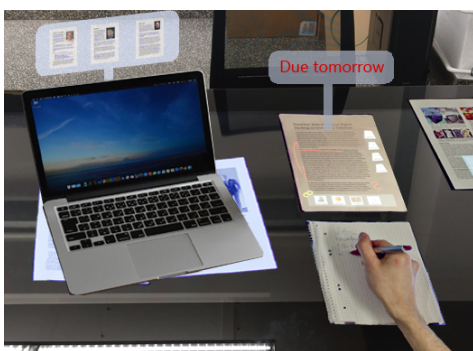


Figure 8.3: Displaying indicative labels for occluded content.

Displaying labels over real-world scenery is one of the common and well-explored standard use cases of AR. However, we combine it with table-tops to leverage the third dimension in complex settings (e.g., occlusion through tall 3D objects) to display labels indicating content hidden behind the object using labels hovering above the content instead of moving them to an unoccluded space on the table (see figure 8.3). Further, we can

leverage the private aspect of the AR display to show additional individualized information for objects (e.g., the actual purchase price for the merchant during a sales talk, while the customer sees his current price or availability

information).

8.3.3 Altering and Adding Object Appearance

Objects (e.g., tangibles) can be interactively configured on digital tabletops. Simple customizations like changing texture can easily be visualized using the *on* and *below layer*, as done by Lindlbauer et al. [2017]. If we add or remove parts, e.g., a spoiler on a car model, the *above layer* can be used to visually add them or blend over them so that they seem to be removed. Further, the texture of an object can be altered by projection. Figure 8.4 shows an example of adding a driver figurine to a 3D-printed formula one car model that is textured via the *on layer*. Further, it is possible to expand content from the screen into the third dimension that is inherently 3D like CAD models or similar things.



Figure 8.4: Altering object appearance by changing color or adding object parts.

8.3.4 3D Hypermovement

In hybrid settings, moving digital content across the tabletop is often difficult since a direct path is blocked. Moving the digital content around all physical objects solely in the *below layer* may be possible, but is very cumbersome. Although 2D techniques have been developed to move digital content across physical obstacles (e.g., hyper dragging [Rekimoto and Saitoh, 1999] or teleport [Khalilbeigi et al., 2013]), they are limited since the moved object is bound to the 2D surface. In the physical world, we grab an object in such cases, lift it, and put it back down at the target location. Using OverTop, this real-world interaction can be easily implemented by augmenting the digital object while being held and moved across the surface (see figure 8.5). Expanding on this

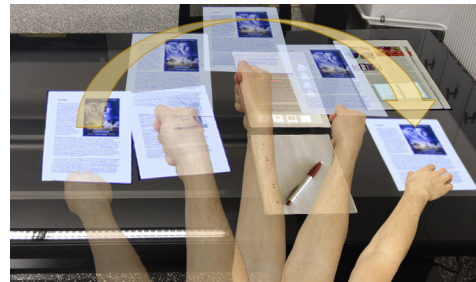


Figure 8.5: 3D Hypermovement allows to naturally move digital content across obstacles.

and keeping it within the scope of the original hyper dragging, this is also possible among different devices (e.g., a laptop screen and the tabletop surface or a smartphone).

8.3.5 3D Clipboard



Figure 8.6: The 3D clipboard is used to store and spatially group content (images and a text fragment) around the workspace.

Digital clipboards are often only usable for single bits of information, that are invisible while being on the clipboard. One cannot always be sure whether a copy operation was successful, leading to surprises when pasting the information. Using OverTop, we propose a three-dimensional clipboard that allows the user to select a piece of information (e.g., contact data, a passage of text, etc.) and actually move it out of the physical or

digital entity containing it into the *above layer*. It is then visualized as a 3D object that can be moved and placed back at a target location to be pasted. In 3D space, one can have and freely place and group multiple objects at the same time while keeping a visual overview (see figure 8.6). At the same time, the clipboard is private since it is displayed on the personal device. One can then select content, modify it in AR, and place it back onto the public surface using 3D Hypermove.

8.4 Summary and Conclusion

In this chapter, we presented OverTop, a 3D tabletop concept for presenting interfaces and content on three layers for interactive tabletop workplaces: *below*, *on*, and *above*. Further, the chapter provided a set of proof-of-concept interaction possibilities for using the three layers in interactive workplace setups.

By leveraging the different display layers with their respective properties, it becomes possible to take a new approach on designing interfaces. Depending on the characteristics of the content to be visualized and interacted with, a suitable layer can be chosen. Thereby, a more efficient use of the available

space as well as a clearer structuring of the interface becomes possible. For instance, instead of displaying information related to a tangible on the tabletop display (below layer), it can now either be visualized on the tangible's surface if the surface is suitable using the on layer, or in the airspace above the tangible using the above layer. Thereby, space on the tabletop display is saved, and the relation of the information to the tangible becomes clearer.

Besides the more efficient use of space, the assignment of content of layers can be made dependent on the content's properties. For inherently three-dimensional data, the above layer is especially suited. The same holds for private information that can be displayed to individual users through their personal AR glasses. Another advantage of the above and on layers is that they can display information at locations on the object rather than only next to an object, as it is the case for the below layer. Thereby, a direct annotation of parts of an object becomes possible. As a result, interaction can also be made more direct since the user can interact directly on the object surface rather than at a remote location on the tabletop. The below layer is especially suited for larger content objects, such as digital documents, or content that is naturally below a physical object, e.g., a contextual map for a tangible.

The presented proof-of-concept interactions can serve as a basis for further research exploring the practical use of the different layers in the design of interaction concepts.

Summary and Conclusion

In the following, we summarize the contributions and main outcomes presented in this thesis grouped along the three main parts II to IV. After the summary, we present directions for future research in chapter 10.

Current practically used hybrid work settings consist of a conventional desk for physical documents and a computer screen used for the digital work environment, as depicted in figure 9.1a. As a result, digital and physical documents are strictly separated. While being a promising solution to integrating the two worlds, tabletops are currently still mostly used for digital documents only (see figure 9.1b). A prominent reason for this is the lacking support for physical objects beyond basic tangibles on tabletops, which causes a set of issues such as, for instance, false touch events or occlusion.

In order to overcome this, the main goal set in the introduction was to foster the integration of digital and physical documents to provide the user with a seamlessly integrated workspace. To reach this goal, we first introduced the technical concept for a hybrid tabletop environment in chapter 2 that leverages the advantages of multiple display technologies to expand the otherwise limited available screen space beyond the table surface to what we called “*peripheral displays*”. At the same time, the combination of an LCD with projector-based top-projection ensures maximum display quality. By means of a depth camera,

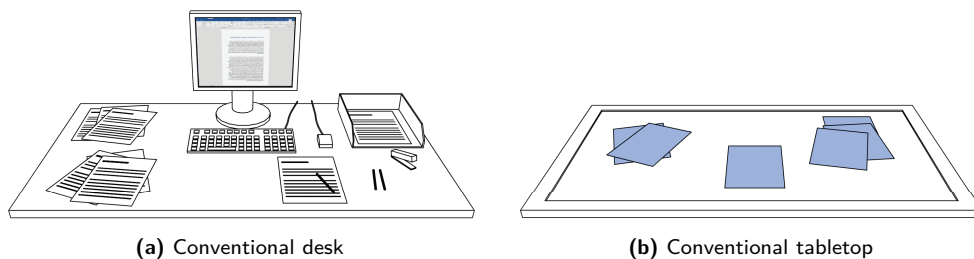


Figure 9.1: Currently, there is a strict separation between office desks (left) used to work with physical documents and digital tabletops (right) that are mainly used with digital documents (depicted in blue). As a result, they are currently two distinct pieces of furniture.

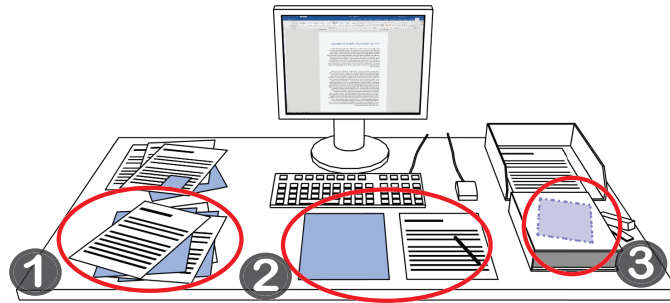


Figure 9.2: An integrated hybrid knowledge work environment providing (1) hybrid stacks where digital (blue) and physical (white) documents can be arbitrarily stacked on top of one another, (2) a surface on which digital and physical documents can coexist next to one another, and (3) support for in-place occlusion mitigation by providing awareness of occluded digital objects leveraging the surface of the occluder.

touch on the peripheral displays is sensed, thereby making them interactive, just like the primary tabletop LCD. This setting served as a basis for the implementation of the following contributions that target more specific issues in hybrid settings.

Stationary Hybrid Knowledge Work With a digital tabletop as a working desk instead of a conventional desk, digital and physical documents can now coexist on the same surface (see figure 9.2 (2)). The most salient problem when using a tabletop as a desk is that physical objects occlude the display, thereby diminishing the space available for interaction as well as hindering the perception of digital content. Past research mainly focused on resolving occlusion by using the primary display of the tabletop to display “proxy” objects. While this provides awareness of occluded objects, it requires additional space on the display beyond what is already occluded by physical objects. On already very cluttered tables, this might lead to further problems. Also, the perception of the spatial layout of documents is hindered even if the proxies are displayed as close as possible to the actual location of the represented object. In order to circumvent these issues, in chapter 3, we contributed a set of visualization and interaction techniques that leverage the peripheral display concept and visualize occluded objects in place and provide direct in situ access to them by making the occluder temporarily “transparent” (see figure 9.2 (3)). A user study showed that this concept is beneficial and outperforms conventional occlusion-awareness strategies that rely on the tabletop surface time and error-wise.

Based on the results of this first contribution, we went one step further and

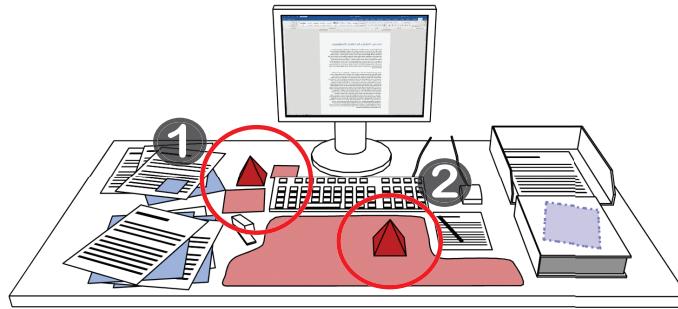


Figure 9.3: For a seamless integration of tangibles and dynamic UIs, (1) an automatic optimization-based placement for digital widgets on cluttered tabletops and (2) a “maximum space”-layout technique are contributed.

investigated concepts for hybrid stacking in chapter 4. While in the previous contribution, digital documents were located either on the tabletop surface or partially on top of occluding objects, they now can be seamlessly intertwined with physical documents (see figure 9.2 (1)). Thereby, the formation of stacks containing both digital and physical documents becomes possible, as easy as it was before with digital- or physical-only stacks. We further provided a set of interaction techniques for easy access to documents within such a hybrid stack. An evaluation confirmed that hybrid stacks are a well-received and easy-to-understand concept that has great potential to facilitate working in hybrid environments.

Layout for Hybrid Environments When projecting onto documents or displaying content in cluttered environments, it is crucial to do so at a suitable position that provides enough space and sufficient display quality (especially in case of projection) so that the user can work with the content in a meaningful way. To that end, we first contributed a continuous projectability measure in chapter 5 and second, based on that measure, an approach for the dynamic generation of optimized layouts in chapter 6.

The projectability measure goes beyond the current state of the art by not only considering binary features like edges or corners or single continuous features like saliency but also combining multiple complementary features. The presented approach considers the surface’s color-based lightness and smoothness, as well as its physical structure by means of a color and depth camera. All these features are measured on a continuous scale and then weighted to calculate the score on a per-pixel base. To allow customization by the end user as well as a flexible adaption to the requirements of a concrete scenario, the weights of the

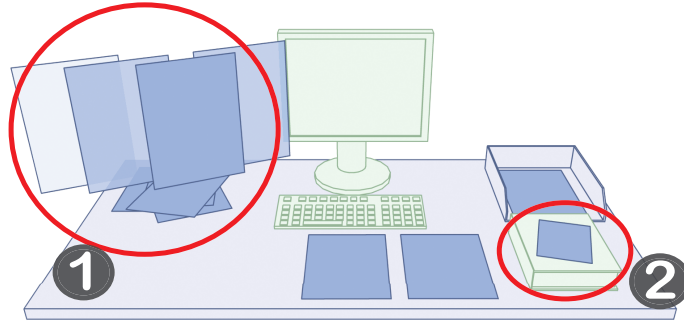


Figure 9.4: A remote view of the hybrid desk in figure 9.2 with (1) an unfolded stack that is opened to access a specific document and (2) an example of a physical landmark (green objects), which is visualized to denote the occlusion state of a document.

different features are customizable. An additional user mask that is factored in as well allows to even further customize the assessment by specifying a “penalty” for specific areas.

To show the applicability of the projectability measure beyond simple layout tasks like finding a static rectangle of a given size that is suitable for display, we further contributed an integrated layout and placement framework that leverages the projectability measure to integrate user interfaces for tangible-based interfaces in hybrid settings. To that end, we contributed a set of layout techniques that, on the one hand, allow dynamically placing widgets while taking into account a set of defined restrictions (see figure 9.3 (1)) and, on the other hand, maximize the space used by a contextual projection by dynamically clipping it around obstacles (see figure 9.3 (2)). Further, we demonstrate an approach for the real-time point-cloud-based tracking of tangibles that are then augmented via projection based on the layout approaches. The complete system is evaluated in a first study that shows its potential and real-time capabilities.

Mobile Workspaces All of the previous contributions mainly focused on stationary use. As a result, there is now the possibility of an integrated hybrid workspace that is conveniently usable in an environment that is equipped with the required equipment. Considering the increasing trend toward mobile work, having such an environment poses a set of new challenges. When leaving the stationary office – e.g., for home office or working at remote locations – the hybrid workspace is not easily taken along. Usually, the physical part is carried in a briefcase, while the digital part is on a laptop or similar personal device. Also, it is often unclear which documents are needed while working

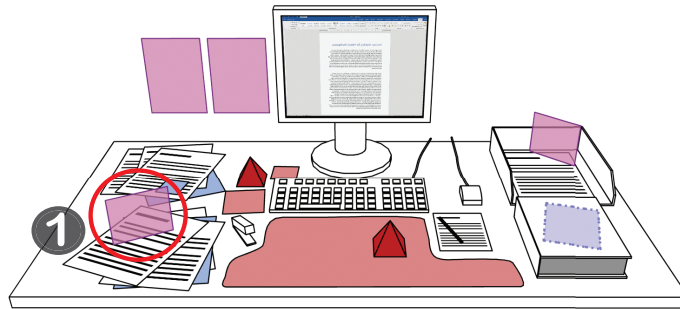


Figure 9.5: A hybrid workspace with additional AR support (objects visualized via AR are shown in purple). Using AR, it becomes possible to expand the interaction and visualization space into the third dimension, e.g., by displaying notifications (1).

from a remote location, and it is not feasible to always carry everything. To overcome these issues, we presented an augmented-reality-based concept for remote access to stationary hybrid work environments that builds on the idea of not only providing full access to the stationary hybrid work environment in chapter 7 (see figure 9.4) but also preserving the useful spatial layout of this environment. For instance, large physical objects are shown as passive proxy objects to provide landmarks (see figure 9.4 (2)). Thereby, the user can leverage his or her spatial memory of the stationary workplace to quickly retrieve documents while on the go. Additionally, a set of interaction and visualization concepts is provided that eases working with the documents; e.g., it is possible to spatially unfold a stack (see figure 9.4 (1)). We further presented the results of two user studies evaluating the mobile system.

General Conclusion Summarizing all of this, this thesis contributed a comprehensive set of interaction techniques and technical approaches to provide a consistent hybrid workspace that allows the user to conveniently use digital and physical content along each other in stationary and mobile settings. Additionally, we presented a first concept for how augmented reality in chapter 8, which we used for mobile access, can be used beneficially as well in stationary environments to further enhance the initial concept of peripheral displays. Thereby, it becomes possible to extend the interaction space into the third dimension (see figure 9.5 (1)). Further, a set of interactions that can benefit from the use of AR was presented.

Future Research Directions

To conclude this thesis, we discuss potential directions for future research in the context of the contributions presented before in the following.

Document Tracking A crucial part of the document-related contributions in this thesis, especially StackTop and PiraTop, is the recognition of specific documents. While for the basic occlusion management in ProjecTop, it is sufficient to detect the presence of occluding objects, the other contributions need to know exactly which document is placed where to build the a model of the stack and consequently the workspace model.

While we implemented a first approach described in chapter 2, it only works for tidily arranged documents. The reason is that in order to identify a specific document, it needs to be isolated within the camera image. With documents starting to overlap arbitrarily, it becomes difficult to find the boundaries of a page or, more generally object, to do so.

Augmented Reality on Tabletops As mentioned in chapter 8, the use of augmented reality in the context of hybrid tabletop environments can set the stage for a multitude of new interactions with documents and data. While some possible interactions leveraging the OverTop concept were already proposed in chapter 8, their practical usability as well as the concrete design of the respective techniques need to be explored. Further, the OverTop concept only gives a first idea of a possible distribution of content among the three layers. The practical use and distribution of content between the layers needs to be studied beyond the initial concept presented. In this context, the use of other modalities – such as spatial audio, which becomes technically possible through the AR headsets – could be explored as well.

Use of Flexible Displays Instead of Paper While we use a two-class display system for the stationary hybrid workspace (high-resolution LCD

tabletop plus top-projection), the respective concepts presented in this thesis directly transfer to other hybrid settings as well:

- LCD-based high-resolution tabletop with passive physical objects and top-projection for document augmentation (our main setup)
- LCD-based tabletop with a set of active e-Ink or similar displays (on-document visualization can be directly displayed)
- Completely top-projected environments (like PeriTop light)

However, especially the use of flexible active displays, such as e-Ink or OLED, opens up a new set of possibilities for interaction. For instance, in the context of hybrid stacking and remote access, it would actually enable swapping the content of physical documents. As a result, some of the interaction techniques described would need to be adapted to the new possibilities.

From Remote Access to Telepresence PiraTop currently provides remote access at the desk level. However, one can imagine going further. The work environment of a knowledge worker usually does not end at the edge of the desk but also comprises furniture such as filing cabinets or shelves that are arranged in an office. The logical next step in this context would be to expand PiraTop’s capabilities to the office (room) level. To that end, a salient question is how to display the whole room in AR given the physical place the user is located. While it is still comparatively easy to place a table, the entire set of office furniture may become challenging and require adaption that preserves the spatial perception. In the same direction, integration of the remote workspace with the current physical workspace in a spatial-layout-preserving way needs to be studied. In the next step, actual virtual telepresence can be enabled, where the user not only has access to his/her stationary workspace but is also augmented back there so that coworkers can personally interact with the remote person.

Collaborative Use All contributions in this thesis are currently considering only individual use at a personal working desk or individual mobile work. However, in daily life, one encounters numerous collaborative situations – for instance, meetings or group projects, where not only a single person but multiple people are working at the same time. This leads to further research questions in the context of the contributions made before.

In the fields of *stationary hybrid knowledge work*, and *layout for hybrid environments*, for instance, all visualization elements are currently oriented toward a single user position. If used together with other people sitting around the desk, the question is how they would be aligned and oriented to provide convenient access. Research has emerged that allows different people to see different content at the same spot on a tabletop [Lissermann et al., 2014]. However, it is unclear how this can be made scalable to many persons and different display technologies.

In the field of *mobile workspaces* multiple questions arise. The first targets non-colocated asynchronous collaboration. To support this scenario, we currently provide the in- and outbox system. It would be interesting to study alternatives that require fewer interaction steps and are more direct, e.g., to allow direct access to other desks but with sophisticated access-management strategies that would need to be explored and studied. Similarly, the use of remote access for shared desks (e.g., meeting desks) is an interesting setting to be explored. This can be done either by representing a physically present desk or even a purely virtual collaborative desk without a physical counterpart. This opens up many issues like access rights if not everybody in a team is allowed to access or modify all documents. To preserve the spatial layout, the documents would have to be visualized in a way that does not destroy the spatial perception but also does not leak confidential information.

Field Studies The contributions in this thesis can form a consistent integrated system but were only prototypically implemented. As a result, there are only individual evaluations in lab settings that did not study the impact of the proposed techniques in the field. Hence, field studies using a sophisticated integrated implementation would be insightful to study the long-term effects of the awareness functions, hybrid stacking usage, and impact of spatial memory on remote work.

List of Figures

1.1	Vannevar Bush's Memex.	5
1.2	Thesis structure, contributions, and their relations.	12
2.1	The different areas on a working desk.	26
2.2	Schematic overview of the main types of tabletop implementations.	31
2.3	Shadowing caused by the user and tall objects.	32
2.4	The two-class display concept of PeriTop.	34
2.5	Envisioned Tabletop Setup for PeriTop	35
2.6	a) The tool palette is top-projected on top of the book. b) An e-mail notification icon is shown on the paper stack.	37
2.7	Peripheral ad hoc off-screen storage of objects (dashed rectangles represent digital objects).	38
2.8	Only the primary working area of the table is equipped with an LCD.	38
2.9	Physical objects used to resolve occlusion.	39
2.10	True hybrid piling.	39
2.11	Texturization of a tangible pyramid.	40
2.12	In-situ resolution of occlusion caused by physical objects.	41
2.13	The structured light pattern used by the Kinect.	41
3.1	Collision of off-object occlusion visualizations.	50
3.2	ProjecTop system visualizing the occluded digital objects.	51
3.3	Questionnaire results by visualization.	60
3.4	Visualization overview.	62
3.5	The Hold-to-Peek gesture in action.	64
3.6	An overview of the SemanticPinch gesture.	65
3.7	An overview of the HyperMove technique.	67
3.8	Initial setup. The dashed objects were only present in high clutter conditions.	68
3.9	Average task duration by interface.	72
3.10	Average number of counting errors per trial by clutter level (left) and by interface (right).	72
3.11	NASA TLX results (the full scale ranges from 0 to 20).	74
3.12	Differently colored outlines to categorize objects.	75
4.1	StackTop facilitates mixed digital/physical (hybrid) stacking on interactive tabletops by providing awareness and means of access to stacked documents.	80

LIST OF FIGURES

4.2	Schematic of a hybrid stack: digital (green), hybrid (blue), and physical (black) documents can be stacked arbitrarily.	80
4.3	Tight coupling visualizations.	87
4.4	Nomenclature overview for the different layers of a hybrid stack.	89
4.5	A physical document with a digital Post-it. The in-place access visualizes the digital documents above the physical as a stack of icons in the free margin of the document.	90
4.6	Loose coupling visualizations.	91
4.7	Path-based stack access.	92
4.8	TLX results. Full scale ranges from 0 to 20.	96
4.9	Results from the AttrakDiff Questionnaire for ProjecTop.	97
5.1	Colorized projectability map of two example scenarios.	107
5.2	Abstract structure of the projectability map calculation.	114
5.3	Comparison of corner detection (left) vs. edge map (right): corner detection does not respond well on an image.	115
5.4	Example of dark areas, not recognized by edge detection.	116
5.5	Physical edges are not always recognized on the color image.	117
5.6	The user mask avoids projection in the mouse area by adding a penalty.	118
5.7	Basic strategies for layout generation.	121
6.1	Usage examples of FlowPut	124
6.2	Overview of the FlowPut environment.	135
6.3	Processing pipeline of the point cloud data: raw (left), pre-processed (center), and clustered/tracked (right).	138
6.4	Layout for UI elements based on environmental constraints (left) and maximum space layout (right). Projection is shown in red.	144
6.5	Interference between projection and the content of a physical document.	144
6.6	Visualization of the element [gray] placement on top of a projectability map [blue=good score, red=bad]. The target position of the elements is the black dot on the physical object [red box] (left). Image captured by the IR camera with and without IR filter (right). The projected content is invisible in the filtered image.	145
6.7	Example of an optimized layout.	146
6.8	Example of a Dilate/Erode iteration.	149
6.9	Two interactive example applications: a tangible relief map (left) used as a tangible 3D display, and a tangible car (right) used as a 3D tangible proxy.	151
6.10	Our lab setup.	153

6.11	General setup with a 40% occluded object to evaluate of the performance of our tracking approach.	156
6.12	Results of the evaluation of the performance of our tracking approach.	157
6.13	Ambiguity of touch points in concave areas.	158
7.1	An example of an augmented reality view of PiraTop.	167
7.2	An overview of the PiraTop system.	177
7.3	A hybrid workspace.	177
7.4	A mobile workspace with PiraTop.	178
7.5	Augmented reality visualization of a physical desk with an in- and outbox, a trash can and documents.	185
7.6	Overview of the visualization techniques developed.	188
7.7	Overview of the document-level features of the mobile interface of PiraTop.	190
7.8	Results for time and error for the different visualizations. . . .	195
7.9	Raw TLX results by visualization type plus/minus standard error.	196
7.10	Results from the AttrakDiff questionnaire for PiraTop.	201
7.11	Visualization distribution by task.	202
7.12	Results of the Likert-scale questionnaire; answer possibilites ranged from -3 to 3.	203
8.1	The three-layered interaction space of OverTop and its general hardware setup.	214
8.2	Ad hoc data visualization for selected content.	218
8.3	Displaying indicative labels for occluded content.	218
8.4	Altering object appearance by changing color or adding object parts.	219
8.5	3D Hypermove allows to naturally move digital content across obstacles.	219
8.6	The 3D clipboard is used to store and spatially group content (images and a text fragment) around the workspace.	220
9.1	The current strict separation between office desks and digital tabletops.	223
9.2	An integrated hybrid knowledge-work environment.	224
9.3	Seamless integration of tangibles and dynamic UIs.	225
9.4	A remote view of the hybrid desk.	226
9.5	A hybrid workspace with additional AR support.	227

List of Tables

1.1	Overview of publications. The publication type is abbreviated as follows: EA – Extended Abstract, WP – Workshop Paper, FP – Full Paper, and J – Journal	13
2.1	Properties of selected augmented desktop systems.	20
2.2	Properties of selected systems that focus on spatial layout.	25
2.3	Summary of important properties of the different object classes.	27
2.4	Summary of properties of physical/digital occlusion.	30
2.5	Summary of important properties of top- and back-projection-based tabletops.	32
2.6	Technologies used in the two PeriTop setups.	36
2.7	Overview of the use case scenarios with regard to the types of objects covered primarily.	37
3.1	Properties of selected systems dealing with occlusion.	57
3.3	Information conveyed by different visualization levels (Low, Mid, and High).	58
3.2	Study design overview.	58
3.4	F-statistics for the influence of the visualization type.	59
3.5	Summary of the interaction techniques included in the ProjecTop interface.	66
3.6	Study design overview.	67
4.1	Properties of selected stacking approaches.	85
4.2	Summary of interaction techniques for hybrid documents included in the StackTop interface.	88
4.3	Summary of the interaction techniques for hybrid stacks included in the StackTop interface.	93
4.4	Study Design Overview	94
5.1	Properties of a representative selection of layout systems.	112
6.1	Properties of a representative selection of touch detection approaches.	129
6.2	Properties of the most related object recognition and tracking systems.	133
6.3	Evaluation overview.	154
6.4	Study design overview.	155
6.5	Study design overview.	157

LIST OF TABLES

6.6	Standard deviations of the reported touch coordinates for different surface geometries in mm.	158
6.7	Study design overview.	159
7.1	Properties of selected systems for mobile work.	175
7.2	Summary of the interaction techniques included in the stationary PiraTop interface.	184
7.3	Summary of the stack visualization techniques included in the mobile PiraTop interface.	189
7.4	Summary of the interaction techniques included in the mobile PiraTop interface.	192
7.5	Study design overview.	193
7.6	Study design overview	198
8.1	Summary of the layer properties.	216

Bibliography

- Takamasa Adachi, Seiya Koura, Fumihisa Shibata, and Asako Kimura. Forearm menu: Using forearm as menu widget on tabletop system. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 333–336, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10 . 1145 / 2512349 . 2512393. URL <http://doi.acm.org/10.1145/2512349.2512393>. (Cited on pages 55 and 57.)
- Anand Agarawala and Ravin Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 1283–1292, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124965. URL <http://doi.acm.org/10.1145/1124772.1124965>. (Cited on pages 23, 25, 82, and 85.)
- A. Agarwal, S. Izadi, M. Chandraker, and A. Blake. High precision multi-touch sensing on surfaces using overhead cameras. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, pages 197–200, Oct 2007. doi: 10 . 1109 / TABLETOP.2007.29. (Cited on page 32.)
- Diane E. Bailey and Nancy B. Kurland. A review of telework research: findings, new directions, and lessons for the study of modern work. *Journal of Organizational Behavior*, 23(4):383–400, 2002. ISSN 1099-1379. doi: 10 . 1002/job.144. URL <http://dx.doi.org/10.1002/job.144>. (Cited on page 169.)
- Deepak Bandyopadhyay, Ramesh Raskar, and Henry Fuchs. Dynamic shader lamps: Painting on movable objects. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, ISAR '01, pages 207–, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1375-1. URL <http://dl.acm.org/citation.cfm?id=582828.881324>. (Cited on pages 130, 131, 132, and 133.)
- Patrick Baudisch and Carl Gutwin. Multiblending: Displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 367–374, New York, NY, USA, 2004.

- ACM. ISBN 1-58113-702-8. doi: 10 . 1145 / 985692 . 985739. URL <http://doi.acm.org/10.1145/985692.985739>. (Cited on pages 52 and 57.)
- Patrick Baudisch, Nathaniel Good, and Paul Stewart. Focus plus context screens: Combining display technology with visualization techniques. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 31–40, New York, NY, USA, 2001. ACM. ISBN 1-58113-438-X. doi: 10 . 1145 / 502348 . 502354. URL <http://doi.acm.org/10.1145/502348.502354>. (Cited on page 35.)
- Dan Bauer, Pierre Fastrez, and Jim Hollan. Computationally-enriched 'piles' for managing digital photo collections. In *Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing*, VLHCC '04, pages 193–195, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7803-8696-5. doi: 10 . 1109 / VLHCC . 2004 . 13. URL <http://dx.doi.org/10.1109/VLHCC.2004.13>. (Cited on pages 83 and 85.)
- Paul Beardsley, Jeroen Van Baar, Ramesh Raskar, and Clifton Forlines. Interaction using a handheld projector. *IEEE Computer Graphics and Applications*, 25(1):39–43, 2005. ISSN 0272-1716. doi: <http://doi.ieeecomputersociety.org/10.1109/MCG.2005.12>. (Cited on pages 171 and 175.)
- Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 101–110, New York, NY, USA, 2001. ACM. ISBN 1-58113-438-X. doi: 10.1145/502348.502363. URL <http://doi.acm.org/10.1145/502348.502363>. (Cited on page 111.)
- Blaine A. Bell and Steven K. Feiner. Dynamic space management for user interfaces. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 239–248, New York, NY, USA, 2000. ACM. ISBN 1-58113-212-3. doi: 10.1145/354401.354790. URL <http://doi.acm.org/10.1145/354401.354790>. (Cited on pages 111 and 112.)
- Olha Bondarenko and Ruud Janssen. Documents at hand: Learning from paper to improve digital technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 121–130, New York, NY, USA, 2005. ACM. ISBN 1-58113-998-5. doi: 10.1145/1054972.1054990.

URL <http://doi.acm.org/10.1145/1054972.1054990>. (Cited on page 22.)

Christophe Bortolaso, Matthew Oskamp, T.C. Nicholas Graham, and Doug Brown. Ormis: A tabletop interface for simulation-based training. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 145–154, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512792. URL <http://doi.acm.org/10.1145/2512349.2512792>. (Cited on page 8.)

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. (Cited on pages 121 and 153.)

Peter Brandl, Jakob Leitner, Thomas Seifried, Michael Haller, Bernard Doray, and Paul To. Occlusion-aware menu design for digital tabletops. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3223–3228, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520461. URL <http://doi.acm.org/10.1145/1520340.1520461>. (Cited on pages 55 and 57.)

Eric Brockmeyer, Ivan Poupyrev, and Scott Hudson. Papillon: Designing curved display surfaces with printed optics. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 457–462, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2502027. URL <http://doi.acm.org/10.1145/2501988.2502027>. (Cited on pages 127 and 129.)

Wilhelm Burger and Mark J. Burge. *Principles of Digital Image Processing: Fundamental Techniques*. Springer Publishing Company, Incorporated, 1 edition, 2009a. ISBN 1848001908, 9781848001909. (Cited on page 149.)

Wilhelm Burger and Mark J. Burge. *Principles of Digital Image Processing: Core Algorithms*. Springer Publishing Company, Incorporated, 1 edition, 2009b. ISBN 1848001940, 9781848001947. (Cited on page 149.)

Vannevar Bush and Jingtao Wang. As we may think. *Atlantic Monthly*, 176: 101–108, 1945. (Cited on page 4.)

Xiang Cao and Ravin Balakrishnan. Interacting with dynamically defined information spaces using a handheld projector and a pen. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*,

- UIST '06, pages 225–234, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166289. URL <http://doi.acm.org/10.1145/1166253.1166289>. (Cited on pages 171 and 175.)
- Xiang Cao, Clifton Forlines, and Ravin Balakrishnan. Multi-user interaction using handheld projectors. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 43–52, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-679-0. doi: 10.1145/1294211.1294220. URL <http://doi.acm.org/10.1145/1294211.1294220>. (Cited on pages 171 and 175.)
- Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. Capstones and zebrawidgets: Sensing stacks of building blocks, dials and sliders on capacitive touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2189–2192, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208371. URL <http://doi.acm.org/10.1145/2207676.2208371>. (Cited on pages 82 and 85.)
- Apoorve Chokshi, Teddy Seyed, Francisco Marinho Rodrigues, and Frank Maurer. eplan multi-surface: A multi-surface environment for emergency response planning exercises. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 219–228, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2587-5. doi: 10.1145/2669485.2669520. URL <http://doi.acm.org/10.1145/2669485.2669520>. (Cited on page 8.)
- Marina Cidota, Stephan Lukosch, Dragos Datcu, and Heide Lukosch. Workspace Awareness in Collaborative AR Using HMDs: A User Study Comparing Audio and Visual Notifications. In *Proceedings of the 7th Augmented Human International Conference 2016*, AH '16, pages 3:1—3:8, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3680-2. doi: 10.1145/2875194.2875204. URL <http://doi.acm.org/10.1145/2875194.2875204>. (Cited on pages 173 and 175.)
- Daniel Cotting and Markus Gross. Interactive environment-aware display bubbles. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, pages 245–254, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166291. URL <http://doi.acm.org/10.1145/1166253.1166291>. (Cited on pages 16, 32, 54, 57, 106, 108, 109, 112, 131, and 133.)

- Kiran Dandekar, Balasundar I Raju, and Mandayam a Srinivasan. 3-D Finite-Element Models of Human and Monkey Fingertips to Investigate the Mechanics of Tactile Sense. *Journal of Biomechanical Engineering*, 125(5):682, 2003. ISSN 01480731. doi: 10.1115/1.1613673. URL <http://biomechanical.asmedigitalcollection.asme.org/article.aspx?articleid=1410767>. (Cited on pages 143 and 158.)
- Chi Tai Dang and Elisabeth André. Usage and recognition of finger orientation for multi-touch tabletop interaction. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, pages 409–426, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23765-2. (Cited on page 33.)
- Sebastian Doeweling, Tarik Tahiri, Philipp Sowinski, Benedikt Schmidt, and Mohammadreza Khalilbeigi. Support for collaborative situation analysis and planning in crisis management teams using interactive tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 273–282, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512823. URL <http://doi.acm.org/10.1145/2512349.2512823>. (Cited on page 8.)
- Sebastian Döweling, Tarik Tahiri, Jan Riemann, and Max Mühlhäuser. Collaborative interaction with geospatial data—a comparison of paper maps, desktop gis and interactive tabletops. In Craig Anslow, Pedro Campos, and Joaquim Jorge, editors, *Collaboration Meets Interactive Spaces*, pages 319–348, Cham, 2016. Springer International Publishing. ISBN 978-3-319-45853-3. doi: 10.1007/978-3-319-45853-3_14. URL https://doi.org/10.1007/978-3-319-45853-3_14. (Cited on page 13.)
- Hal Eden, Eric Scharff, and Eva Hornecker. Multilevel design and role play: Experiences in assessing support for neighborhood participation in design. In *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '02, pages 387–392, New York, NY, USA, 2002. ACM. ISBN 1-58113-515-7. doi: 10.1145/778712.778768. URL <http://doi.acm.org/10.1145/778712.778768>. (Cited on page 8.)
- Marge Eldridge, Mik Lamming, Mike Flynn, Chris Jones, and David Pendlebury. Studies of mobile document work and their contributions to the satchel

- project. *Personal Technologies*, 4(2):102–112, 2000. ISSN 1617-4917. doi: 10.1007/BF01324117. URL <http://dx.doi.org/10.1007/BF01324117>. (Cited on page 169.)
- Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. Spatial constancy of surface-embedded layouts across multiple environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, pages 65–68, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3703-8. doi: 10.1145/2788940.2788954. URL <http://doi.acm.org/10.1145/2788940.2788954>. (Cited on pages 111, 112, 132, 133, 166, 173, and 175.)
- K. M. Everitt, M. R. Morris, A. J. B. Brush, and A. D. Wilson. Docudesk: An interactive surface for creating and rehydrating many-to-many linkages among paper and digital documents. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 25–28, Oct 2008. doi: 10.1109/TABLETOP.2008.4660179. (Cited on page 88.)
- Kathryn L. Fonner and Michael E. Roloff. Why teleworkers are more satisfied with their jobs than are office-based workers: When less contact is beneficial. *Journal of Applied Communication Research*, 38(4):336–361, 2010. doi: 10.1080/00909882.2010.513998. URL <http://dx.doi.org/10.1080/00909882.2010.513998>. (Cited on page 169.)
- Euan Freeman and Stephen Brewster. Messy tabletops: Clearing up the occlusion problem. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 1515–1520, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1952-2. doi: 10.1145/2468356.2468627. URL <http://doi.acm.org/10.1145/2468356.2468627>. (Cited on pages 16, 54, 110, and 112.)
- Kensaku Fujii, Jun Shimamura, Kenichi Arakawa, and Tomohiko Arikawa. Tangible search for stacked objects. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 848–849, New York, NY, USA, 2003. ACM. ISBN 1-58113-637-4. doi: 10.1145/765891.766029. URL <http://doi.acm.org/10.1145/765891.766029>. (Cited on pages 80, 82, and 85.)
- Kensaku Fujii, Michael D. Grossberg, and Shree K. Nayar. A projector-camera system with real-time photometric adaptation for dynamic environments. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1180 vol. 2–, Washington,

DC, USA, June 2005. IEEE Computer Society. doi: 10.1109/CVPR.2005.42. (Cited on page 108.)

Markus Funk, Oliver Korn, and Albrecht Schmidt. An augmented workplace for enabling user-defined tangibles. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 1285–1290, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2474-8. doi: 10.1145/2559206.2581142. URL <http://doi.acm.org/10.1145/2559206.2581142>. (Cited on pages 110 and 112.)

Genki Furumi, Daisuke Sakamoto, and Takeo Igarashi. Snaprail: a tabletop user interface widget for addressing occlusion by physical objects. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ITS '12, pages 193–196, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396666. URL <http://doi.acm.org/10.1145/2396636.2396666>. (Cited on pages 16, 55, 57, 63, 109, and 112.)

Ravi S Gajendran and David A Harrison. The good, the bad, and the unknown about telecommuting: meta-analysis of psychological mediators and individual consequences. *Journal of Applied Psychology*, 92(6):1524, 2007. (Cited on pages 9, 166, and 169.)

R. Gal, L. Shapira, E. Ofek, and P. Kohli. Flare: Fast layout for augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 207–212, Sept 2014. doi: 10.1109/ISMAR.2014.6948429. (Cited on pages 111 and 112.)

Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*, pages 449–459, New York, New York, USA, oct 2014a. ACM Press. ISBN 9781450330695. doi: 10.1145/2642918.2647372. URL <http://dl.acm.org/citation.cfm?id=2642918.2647372>. (Cited on pages 173 and 175.)

Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. In Touch with the Remote World: Remote Collaboration with Augmented Reality Drawings and Virtual Navigation. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pages 197–205, New York, NY, USA, 2014b. ACM. ISBN 978-1-4503-3253-8. doi:

- 10.1145/2671015.2671016. URL <http://doi.acm.org/10.1145/2671015.2671016>. (Cited on page 173.)
- Thomas Geymayer, Markus Steinberger, Alexander Lex, Marc Streit, and Dieter Schmalstieg. Show me the invisible: Visualizing hidden content. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3705–3714, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557032. URL <http://doi.acm.org/10.1145/2556288.2557032>. (Cited on pages 53 and 57.)
- Audrey Girouard, Aneesh Tarun, and Roel Vertegaal. Displaystacks: Interaction techniques for stacks of flexible thin-film displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2431–2440, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208406. URL <http://doi.acm.org/10.1145/2207676.2208406>. (Cited on pages 84 and 85.)
- Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. Printsense: A versatile sensing technique to support multimodal flexible surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1407–1410, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557173. URL <http://doi.acm.org/10.1145/2556288.2557173>. (Cited on page 127.)
- J. M. Gottfried, R. Nair, S. Meister, C. S. Garbe, and D. Kondermann. Time of flight motion compensation revisited. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5861–5865, Oct 2014. doi: 10.1109/ICIP.2014.7026184. (Cited on page 161.)
- Raphael Grasset, Tobias Langlotz, Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. Image-driven view management for augmented reality browsers. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '12, pages 177–186, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4673-4660-3. doi: 10.1109/ISMAR.2012.6402555. URL <http://dx.doi.org/10.1109/ISMAR.2012.6402555>. (Cited on pages 110, 112, and 115.)
- M.D. Grossberg, H. Peri, S.K. Nayar, and P.N. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system.

In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–452–I–459 Vol.1, Washington, DC, USA, June 2004. IEEE Computer Society. doi: 10.1109/CVPR.2004.1315067. (Cited on page 108.)

Thomas Guillod. Interpolations, courbes de bézier et b-splines. *Bulletin de la Société des enseignants Neuchâtelois de Sciences*, 2008. (Cited on page 150.)

Sebastian Günther, Martin Schmitz, Florian Müller, Jan Riemann, and Max Mühlhäuser. Byo*: Utilizing 3d printed tangible tools for interaction on interactive surfaces. In *Proceedings of the 2017 ACM Workshop on Interacting with Smart Objects*, SmartObject '17, pages 21–26, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4902-4. doi: 10.1145/3038450.3038456. URL <http://doi.acm.org/10.1145/3038450.3038456>. (Cited on page 13.)

Sebastian Günther, Florian Müller, Martin Schmitz, Jan Riemann, Niloofar Dezfuli, Markus Funk, Dominik Schön, and Max Mühlhäuser. Checkmate: Exploring a tangible augmented reality interface for remote interaction. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, pages LBW570:1–LBW570:6, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5621-3. doi: 10.1145/3170427.3188647. URL <http://doi.acm.org/10.1145/3170427.3188647>. (Cited on page 13.)

Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 115–118, New York, NY, USA, 2005. ACM. ISBN 1-59593-271-2. doi: 10.1145/1095034.1095054. URL <http://doi.acm.org/10.1145/1095034.1095054>. (Cited on page 33.)

Mark Hancock, Otmar Hilliges, Christopher Collins, Dominikus Baur, and Sheelagh Carpendale. Exploring tangible and direct touch interfaces for manipulating 2d and 3d information on a digital table. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 77–84, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731921. URL <http://doi.acm.org/10.1145/1731903.1731921>. (Cited on page 7.)

John Hardy. Experiences: a year in the life of an interactive desk. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 679–688,

- New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1210-3. doi: 10.1145/2317956.2318058. URL <http://doi.acm.org/10.1145/2317956.2318058>. (Cited on page 32.)
- John Hardy and Jason Alexander. Toolkit support for interactive projected displays. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 42:1–42:10, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1815-0. doi: 10.1145/2406367.2406419. URL <http://doi.acm.org/10.1145/2406367.2406419>. (Cited on page 150.)
- Beverly L. Harrison and Kim J. Vicente. An experimental evaluation of transparent menu usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 391–398, New York, NY, USA, 1996. ACM. ISBN 0-89791-777-4. doi: 10.1145/238386.238583. URL <http://doi.acm.org/10.1145/238386.238583>. (Cited on page 52.)
- Beverly L. Harrison, Gordon Kurtenbach, and Kim J. Vicente. An experimental evaluation of transparent user interface tools and information content. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*, UIST '95, pages 81–90, New York, NY, USA, 1995. ACM. ISBN 0-89791-709-X. doi: 10.1145/215585.215669. URL <http://doi.acm.org/10.1145/215585.215669>. (Cited on pages 52 and 57.)
- Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 441–450, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047255. URL <http://doi.acm.org/10.1145/2047196.2047255>. (Cited on pages 166, 172, and 175.)
- Björn Hartmann, Meredith Ringel Morris, Hrvoje Benko, and Andrew D. Wilson. Pictionary: Supporting collaborative design work by integrating physical and digital artifacts. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, pages 421–424, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-795-0. doi: 10.1145/1718918.1718989. URL <http://doi.acm.org/10.1145/1718918.1718989>. (Cited on pages 18, 20, and 106.)
- Marc Hassenzahl, Michael Burmester, and Franz Koller. Attrakdiff: Ein fragebogen zur messung wahrgenommener hedonischer und pragmatischer

qualität. In G. Szwillus and J. Ziegler, editors, *Mensch & Computer 2003: Interaktion in Bewegung*, pages 187–196, Stuttgart, 2003. B. G. Teubner. (Cited on page 200.)

E.Jeffrey Hill, Maria Ferris, and Vjollca Mårtinson. Does it matter where you work? a comparison of how three work venues (traditional office, virtual office, and home office) influence aspects of work and personal/family life. *Journal of Vocational Behavior*, 63(2):220 – 241, 2003. ISSN 0001-8791. doi: [http://dx.doi.org/10.1016/S0001-8791\(03\)00042-3](http://dx.doi.org/10.1016/S0001-8791(03)00042-3). URL <http://www.sciencedirect.com/science/article/pii/S0001879103000423>. Special Issue on Technology and Careers. (Cited on pages 9, 166, and 169.)

Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1063–1072, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557130. URL <http://doi.acm.org/10.1145/2556288.2557130>. (Cited on page 208.)

David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. Paper windows: Interaction techniques for digital paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 591–599, New York, NY, USA, 2005. ACM. ISBN 1-58113-998-5. doi: 10.1145/1054972.1055054. URL <http://doi.acm.org/10.1145/1054972.1055054>. (Cited on pages 84 and 85.)

Jochen Huber, Jürgen Steimle, Chunyuan Liao, Qiong Liu, and Max Mühlhäuser. Lightbeam: Interacting with augmented real-world objects in pico projections. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 16:1–16:10, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1815-0. doi: 10.1145/2406367.2406388. URL <http://doi.acm.org/10.1145/2406367.2406388>. (Cited on pages 106, 107, 171, and 175.)

Edward W. Ishak and Steven K. Feiner. Interacting with hidden content using content-aware free-space transparency. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 189–192, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029666. URL <http://doi.acm.org/10.1145/1029632.1029666>. (Cited on pages 52 and 57.)

- Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258715. URL <http://doi.acm.org/10.1145/258549.258715>. (Cited on pages 7 and 124.)
- Daisuke Iwai and Kosuke Sato. Limpid desk: see-through access to disorderly desktop in projection-based mixed reality. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '06, pages 112–115, New York, NY, USA, 2006. ACM. ISBN 1-59593-321-2. doi: 10.1145/1180495.1180519. URL <http://doi.acm.org/10.1145/1180495.1180519>. (Cited on pages 54, 57, 80, 82, and 85.)
- Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time dynamic 3d surface reconstruction and interaction. In *ACM SIGGRAPH 2011 Talks*, SIGGRAPH '11, pages 23:1–23:1, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0974-5. doi: 10.1145/2037826.2037857. URL <http://doi.acm.org/10.1145/2037826.2037857>. (Cited on page 111.)
- Mikkel Rønne Jakobsen and Kasper Hornbæk. Piles, tabs and overlaps in navigation among documents. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 246–255, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-934-3. doi: 10.1145/1868914.1868945. URL <http://doi.acm.org/10.1145/1868914.1868945>. (Cited on pages 83 and 85.)
- Waqas Javed, KyungTae Kim, Sohaib Ghani, and Niklas Elmqvist. Evaluating physical/virtual occlusion management techniques for horizontal displays. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction - INTERACT 2011*, volume 6948 of *Lecture Notes in Computer Science*, pages 391–408. Springer Berlin Heidelberg, Heidelberg, Germany, 2011. ISBN 978-3-642-23764-5. doi: 10.1007/978-3-642-23765-2_27. URL http://dx.doi.org/10.1007/978-3-642-23765-2_27. (Cited on pages 16, 39, 55, 57, 58, 63, 67, and 68.)
- Ian Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.

1007 / 978-3-642-04898-2 _ 455. URL https://doi.org/10.1007/978-3-642-04898-2_455. (Cited on page 139.)

B.R. Jones, R. Sodhi, R.H. Campbell, G. Garnett, and B.P. Bailey. Build your world and play in it: Interacting with surface particles on complex objects. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 165–174, Washington, DC, USA, 2010. IEEE Computer Society. doi: 10.1109/ISMAR.2010.5643566. (Cited on page 110.)

Sasa Junuzovic, Kori Inkpen, Tom Blank, and Anoop Gupta. Illumishare: Sharing any surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1919–1928, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208333. URL <http://doi.acm.org/10.1145/2207676.2208333>. (Cited on page 146.)

H. Kajimoto, N. Kawakami, S. Tachi, and M. Inami. Smarttouch: electric skin to touch the untouchable. *IEEE Computer Graphics and Applications*, 24 (1):36–43, Jan 2004. ISSN 0272-1716. doi: 10.1109/MCG.2004.1255807. (Cited on page 208.)

Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, Beverly Harrison, Adam D. Rea, Matthai Philipose, and Anthony LaMarca. Bonfire: A nomadic system for hybrid laptop-tabletop interaction. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 129–138, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-745-5. doi: 10.1145/1622176.1622202. URL <http://doi.acm.org/10.1145/1622176.1622202>. (Cited on pages 32, 35, 107, 171, and 175.)

Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. Instant inkjet circuits: Lab-based inkjet printing to support rapid prototyping of ubicomp devices. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 363–372, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1770-2. doi: 10.1145/2493432.2493486. URL <http://doi.acm.org/10.1145/2493432.2493486>. (Cited on page 127.)

Mohammadreza Khalilbeigi, Jürgen Steimle, and Max Mühlhäuser. Interaction techniques for hybrid piles of documents on interactive tabletops. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3943–3948, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-930-5.

- doi: 10.1145/1753846.1754083. URL <http://doi.acm.org/10.1145/1753846.1754083>. (Cited on pages 13, 54, 80, 83, and 85.)
- Mohammadreza Khalilbeigi, Roman Lissermann, Wolfgang Kleine, and Jürgen Steimle. Foldme: Interacting with double-sided foldable displays. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 33–40, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1174-8. doi: 10.1145/2148131.2148142. URL <http://doi.acm.org/10.1145/2148131.2148142>. (Cited on page 110.)
- Mohammadreza Khalilbeigi, Jürgen Steimle, Jan Riemann, Niloofar Dezfuli, Max Mühlhäuser, and James D. Hollan. Objectop: occlusion awareness of physical objects on interactive tabletops. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, ITS '13, pages 255–264, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512806. URL <http://doi.acm.org/10.1145/2512349.2512806>. (Cited on pages 13, 16, 55, 57, 58, 62, 63, 65, 66, 67, 68, 88, 109, 112, and 219.)
- Ahmed Kharrufa, Madeline Balaam, Phil Heslop, David Leat, Paul Dolan, and Patrick Olivier. Tables in the wild: Lessons learned from a large-scale multi-tabletop deployment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1021–1030, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466130. URL <http://doi.acm.org/10.1145/2470654.2466130>. (Cited on page 6.)
- Alison Kidd. The marks are on the knowledge worker. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 186–191, New York, NY, USA, 1994. ACM. ISBN 0-89791-650-6. doi: 10.1145/191666.191740. URL <http://doi.acm.org/10.1145/191666.191740>. (Cited on page 4.)
- David Kim, Shahram Izadi, Jakub Dostal, Christoph Rhemann, Cem Keskin, Christopher Zach, Jamie Shotton, Timothy Large, Steven Bathiche, Matthias Nießner, D. Alex Butler, Sean Fanello, and Vivek Pradeep. Retrodepth: 3d silhouette sensing for high-precision input on and above physical surfaces. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 1377–1386, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557336. URL <http://doi.acm.org/10.1145/2556288.2557336>. (Cited on page 32.)

- Han-Jong Kim, Ju-Whan Kim, and Tek-Jin Nam. ministudio: Designers' tool for prototyping ubicomp space with interactive miniature. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 213–224, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858180. URL <http://doi.acm.org/10.1145/2858036.2858180>. (Cited on pages 131 and 133.)
- Jiwon Kim, Steven M. Seitz, and Maneesh Agrawala. Video-based document tracking: Unifying your physical and electronic desktops. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 99–107, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029650. URL <http://doi.acm.org/10.1145/1029632.1029650>. (Cited on pages 18, 20, 23, and 25.)
- David Kirk, Abigail Sellen, Stuart Taylor, Nicolas Villar, and Shahram Izadi. Putting the physical into the digital: Issues in designing hybrid interactive surfaces. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 35–44, Swinton, UK, UK, 2009. British Computer Society. URL <http://dl.acm.org/citation.cfm?id=1671011.1671016>. (Cited on page 7.)
- David Kirk, Shahram Izadi, Otmar Hilliges, Richard Banks, Stuart Taylor, and Abigail Sellen. At home with surface computing? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 159–168, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207699. URL <http://doi.acm.org/10.1145/2207676.2207699>. (Cited on page 6.)
- David Kirsh. The intelligent use of space. *Artif. Intell.*, 73(1-2):31–68, February 1995. ISSN 0004-3702. doi: 10.1016/0004-3702(94)00017-U. URL [http://dx.doi.org/10.1016/0004-3702\(94\)00017-U](http://dx.doi.org/10.1016/0004-3702(94)00017-U). (Cited on pages 4, 21, and 207.)
- David Kirsh. The context of work. *Hum.-Comput. Interact.*, 16(2):305–322, December 2001. ISSN 0737-0024. doi: 10.1207/S15327051HCI16234_12. URL http://dx.doi.org/10.1207/S15327051HCI16234_12. (Cited on page 21.)
- Konstantin Klamka and Raimund Dachzelt. Illumipaper: Illuminated interactive paper. In *Proceedings of the 2017 CHI Conference on Human Factors*

- in Computing Systems*, CHI '17, pages 5605–5618, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025525. URL <http://doi.acm.org/10.1145/3025453.3025525>. (Cited on pages 17 and 20.)
- Florian Klompmaier, Karsten Nebe, and Alex Fast. dsensingni: A framework for advanced tangible interaction using a depth camera. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 217–224, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1174-8. doi: 10.1145/2148131.2148179. URL <http://doi.acm.org/10.1145/2148131.2148179>. (Cited on pages 40, 128, and 129.)
- Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998. ISBN 0-201-89685-0. (Cited on page 142.)
- Daisuke Kobayashi and Naoki Hashimoto. Spatial augmented reality by using depth-based object tracking. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 33:1–33:1, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2958-3. doi: 10.1145/2614217.2614226. URL <http://doi.acm.org/10.1145/2614217.2614226>. (Cited on pages 132 and 139.)
- Kazue Kobayashi, Atsunobu Narita, Mitsunori Hirano, Ichiro Kase, Shinetsu Tsuchida, Takaharu Omi, Tatsuhito Kakizaki, and Takuma Hosokawa. Collaborative simulation interface for planning disaster measures. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 977–982, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: 10.1145/1125451.1125639. URL <http://doi.acm.org/10.1145/1125451.1125639>. (Cited on page 8.)
- H. Koike, Y. Sato, and Y. Kobayashi. Integrating paper and digital information on enhanceddesk: a method for realtime finger tracking on an augmented desk system. *ACM Trans. Comput.-Hum. Interact.*, 8(4):307–322, December 2001. ISSN 1073-0516. doi: 10.1145/504704.504706. URL <http://doi.acm.org/10.1145/504704.504706>. (Cited on pages 17, 128, 129, 131, 133, and 169.)
- Ryo Koizumi, Daisuke Kobayashi, and Naoki Hashimoto. Acceleration of dynamic spatial augmented reality system with a depth camera. In *Proceedings of the 2015 International Conference on Cyberworlds (CW)*, CW

- '15, pages 50–53, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-9403-1. doi: 10.1109 / CW.2015.42. URL <http://dx.doi.org/10.1109 / CW.2015.42>. (Cited on page 132.)
- Daisuke Kondo, Yuichi Shiwa, and Ryugo Kijima. Free form projection display and application. In *Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems*, PROCAMS '08, pages 8:1–8:2, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-272-6. doi: 10.1145/1394622.1394633. URL <http://doi.acm.org/10.1145/1394622.1394633>. (Cited on page 131.)
- Seiya Koura, Shunsuke Suo, Asako Kimura, Fumihisa Shibata, and Hideyuki Tamura. Amazing forearm as an innovative interaction device and data storage on tabletop display. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 383–386, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396706. URL <http://doi.acm.org/10.1145/2396636.2396706>. (Cited on page 55.)
- Hideaki Kuzuoka. Spatial workspace collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, pages 533–540, New York, New York, USA, jun 1992. ACM Press. ISBN 0897915135. doi: 10.1145/142750.142980. URL <http://dl.acm.org/citation.cfm?id=142750.142980>. (Cited on pages 173 and 175.)
- Mik Lamming, Marge Eldridge, Mike Flynn, Chris Jones, and David Pendlebury. Satchel: Providing access to any document, any time, anywhere. *ACM Trans. Comput.-Hum. Interact.*, 7(3):322–352, September 2000. ISSN 1073-0516. doi: 10.1145/355324.355326. URL <http://doi.acm.org/10.1145/355324.355326>. (Cited on pages 172 and 175.)
- D. Leithinger and M. Haller. Improving menu interaction for cluttered tabletop setups with user-drawn path menus. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, pages 121–128, Washington, DC, USA, Oct 2007. IEEE Computer Society. doi: 10.1109/TABLETOP.2007.24. (Cited on pages 54, 57, and 109.)
- Chunyuan Liao, Hao Tang, Qiong Liu, Patrick Chiu, and Francine Chen. Fact: fine-grained cross-media interaction with documents via a portable hybrid

- paper-laptop interface. In *Proceedings of the international conference on Multimedia*, MM '10, pages 361–370, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874001. URL <http://doi.acm.org/10.1145/1873951.1874001>. (Cited on pages 17, 20, 106, 108, 109, 112, 115, 131, and 133.)
- David Lindlbauer, Jörg Mueller, and Marc Alexa. Changing the appearance of real-world objects by modifying their surroundings. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3954–3965, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025795. URL <http://doi.acm.org/10.1145/3025453.3025795>. (Cited on pages 110 and 219.)
- Roman Lissermann, Jochen Huber, Martin Schmitz, Jürgen Steimle, and Max Mühlhäuser. Permulin: Mixed-focus collaboration on multi-view tabletops. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3191–3200, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557405. URL <http://doi.acm.org/10.1145/2556288.2557405>. (Cited on pages 216 and 231.)
- David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL <https://doi.org/10.1023/B:VISI.0000029664.99615.94>. (Cited on page 43.)
- Paul Luff and Christian Heath. Mobility in collaboration. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 305–314, New York, NY, USA, 1998. ACM. ISBN 1-58113-009-0. doi: 10.1145/289444.289505. URL <http://doi.acm.org/10.1145/289444.289505>. (Cited on page 6.)
- Paul Luff, Christian Heath, and David Greatbatch. Tasks-in-interaction: Paper and screen based documentation in collaborative activity. In *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, CSCW '92, pages 163–170, New York, NY, USA, 1992. ACM. ISBN 0-89791-542-9. doi: 10.1145/143457.143475. URL <http://doi.acm.org/10.1145/143457.143475>. (Cited on pages 6, 169, and 172.)
- Thomas W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112,

January 1983. ISSN 1046-8188. doi: 10.1145/357423.357430. URL <http://doi.acm.org/10.1145/357423.357430>. (Cited on pages 4, 21, 80, 86, 187, and 207.)

Richard Mander, Gitta Salomon, and Yin Yin Wong. A “pile” metaphor for supporting casual organization of information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 627–634, New York, NY, USA, 1992. ACM. ISBN 0-89791-513-5. doi: 10.1145/142750.143055. URL <http://doi.acm.org/10.1145/142750.143055>. (Cited on pages 21, 82, and 85.)

Jean M. Mandler, Dale Seegmiller, and Jeanne Day. On the coding of spatial information. *Memory & Cognition*, 5(1):10–16, 1977. ISSN 1532-5946. doi: 10.3758/BF03209185. URL <http://dx.doi.org/10.3758/BF03209185>. (Cited on pages 4 and 22.)

Steve Mann. Telepointer: Hands-free completely self-contained wearable visual augmented reality without headwear and without any infrastructural reliance. In *Wearable Computers, The Fourth International Symposium on*, pages 177–178, Los Alamitos, CA, USA, Oct 2000. IEEE Computer Society. doi: 10.1109/ISWC.2000.888489. (Cited on pages 172 and 175.)

Steve Mann, James Fung, and Eric Moncrieff. Eyetap technology for wireless electronic news gathering. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(4): 19–26, October 1999. ISSN 1559-1662. doi: 10.1145/584039.584044. URL <http://doi.acm.org/10.1145/584039.584044>. (Cited on page 172.)

Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw - wear ur world: A wearable gestural interface. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4111–4116, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520626. URL <http://doi.acm.org/10.1145/1520340.1520626>. (Cited on pages 166, 172, and 175.)

H. Mitsuhashi, Y. Yano, and T. Moriyama. Paper-top interface for supporting note-taking and its preliminary experiment. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 3456–3462, Washington, DC, USA, Oct 2010. IEEE Computer Society. doi: 10.1109/ICSMC.2010.5642448. (Cited on pages 17, 20, 108, 112, 131, and 133.)

Max Mühlhäuser, Mohammadreza Khalilbeigi, Jan Riemann, Sebastian Döweling, and Roman Lissermann. Immersive tabletop: der digitale tisch als tisch. *Informatik-Spektrum*, 37(5):407–412, Oct 2014. ISSN 1432-122X. doi: 10.1007/s00287-014-0799-z. URL <https://doi.org/10.1007/s00287-014-0799-z>. (Cited on page 13.)

Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1235–1248, March 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2592910. URL <https://doi.org/10.1109/TVCG.2016.2592910>. (Cited on page 131.)

NASA Ames Research Center, Moffet Field. Nasa tlx, 1988. (Cited on page 70.)

Moshe Naveh-Benjamin. Coding of spatial location information: An automatic process?. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(4):595 – 605, 1987. ISSN 0278-7393. URL <http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1988-03384-001&site=ehost-live>. (Cited on pages 4, 22, and 23.)

Shree K. Nayar, Harish Peri, Michael D. Grossberg, and Peter N. Belhumeur. A projection system with radiometric compensation for screen imperfections. In *IEEE ICCV Workshop on Projector-Camera Systems (PROCAMS)*, Washington, DC, USA, 2003. IEEE Computer Society. (Cited on page 108.)

Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. Virtual Replicas for Remote Assistance in Virtual and Augmented Reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 405–415, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3779-3. doi: 10.1145/2807442.2807497. URL <http://doi.acm.org/10.1145/2807442.2807497>. (Cited on pages 173 and 175.)

Masa Ogata and Masaaki Fukumoto. Fluxpaper: Reinventing paper with dynamic actuation powered by magnetic flux. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 29–38, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702516. URL <http://doi.acm.org/10.1145/2702123.2702516>. (Cited on page 183.)

- Kenton O'Hara and Abigail Sellen. A comparison of reading paper and on-line documents. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 335–342, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258787. URL <http://doi.acm.org/10.1145/258549.258787>. (Cited on page 3.)
- Simon Olberding, Nan-Wei Gong, John Tiab, Joseph A. Paradiso, and Jürgen Steimle. A cuttable multi-touch sensor. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 245–254, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2502048. URL <http://doi.acm.org/10.1145/2501988.2502048>. (Cited on pages 127 and 129.)
- Simon Olberding, Michael Wessely, and Jürgen Steimle. Printscreen: Fabricating highly customizable thin-film touch-displays. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 281–290, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3069-5. doi: 10.1145/2642918.2647413. URL <http://doi.acm.org/10.1145/2642918.2647413>. (Cited on page 127.)
- Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. Touch & activate: Adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 31–40, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2501989. URL <http://doi.acm.org/10.1145/2501988.2501989>. (Cited on pages 127 and 129.)
- Jason Orlosky, Kiyoshi Kiyokawa, and Haruo Takemura. Dynamic text management for see-through wearable and heads-up display systems. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 363–370, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1965-2. doi: 10.1145/2449396.2449443. URL <http://doi.acm.org/10.1145/2449396.2449443>. (Cited on page 110.)
- Gunhyuk Park, Seungmoon Choi, Kyunghun Hwang, Sunwook Kim, Jaechon Sa, and Moonchae Joung. Tactile effect design and evaluation for virtual buttons on a mobile device touchscreen. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 11–20, New York, NY, USA, 2011.

- ACM. ISBN 978-1-4503-0541-9. doi: 10.1145/2037373.2037376. URL <http://doi.acm.org/10.1145/2037373.2037376>. (Cited on page 208.)
- James Patten and Hiroshi Ishii. A comparison of spatial organization strategies in graphical and tangible user interfaces. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, pages 41–50, New York, NY, USA, 2000. ACM. doi: 10.1145/354666.354671. URL <http://doi.acm.org/10.1145/354666.354671>. (Cited on page 22.)
- Mark Perry, Kenton O'hara, Abigail Sellen, Barry Brown, and Richard Harper. Dealing with mobility: Understanding access anytime, anywhere. *ACM Trans. Comput.-Hum. Interact.*, 8(4):323–347, December 2001. ISSN 1073-0516. doi: 10.1145/504704.504707. URL <http://doi.acm.org/10.1145/504704.504707>. (Cited on page 172.)
- Anne Marie Piper and James D. Hollan. Tabletop displays for small group study: Affordances of paper and digital materials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1227–1236, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518885. URL <http://doi.acm.org/10.1145/1518701.1518885>. (Cited on page 7.)
- Thomas Porter and Tom Duff. Compositing digital images. *SIGGRAPH Comput. Graph.*, 18(3):253–259, January 1984. ISSN 0097-8930. doi: 10.1145/964965.808606. URL <http://doi.acm.org/10.1145/964965.808606>. (Cited on page 53.)
- Kathrin Probst, Michael Haller, Kentaro Yasu, Maki Sugimoto, and Masahiko Inami. Move-it sticky notes providing active physical feedback through motion. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '14, pages 29–36, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2635-3. doi: 10.1145/2540930.2540932. URL <http://doi.acm.org/10.1145/2540930.2540932>. (Cited on page 183.)
- Pasi Pyöriä. The concept of knowledge work revisited. *Journal of Knowledge Management*, 9(3):116–127, 2005. doi: 10.1108/13673270510602818. URL <https://doi.org/10.1108/13673270510602818>. (Cited on page 3.)
- Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. Huddlelamp: Spatially-aware mobile displays for ad-hoc

around-the-table collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, pages 45–54, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2587-5. doi: 10.1145/2669485.2669500. URL <http://doi.acm.org/10.1145/2669485.2669500>. (Cited on page 35.)

Gonzalo Ramos, George Robertson, Mary Czerwinski, Desney Tan, Patrick Baudisch, Ken Hinckley, and Maneesh Agrawala. Tumble! splat! helping users access and manipulate occluded content in 2d drawings. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 428–435, New York, NY, USA, 2006. ACM. ISBN 1-59593-353-0. doi: 10.1145/1133265.1133351. URL <http://doi.acm.org/10.1145/1133265.1133351>. (Cited on pages 53 and 57.)

Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 179–188, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi: 10.1145/280814.280861. URL <http://doi.acm.org/10.1145/280814.280861>. (Cited on pages 18 and 20.)

Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In StevenJ. Gortler and Karol Myszkowski, editors, *Rendering Techniques 2001*, Eurographics, pages 89–102. Springer Vienna, Vienna, Austria, 2001. ISBN 978-3-211-83709-2. doi: 10.1007/978-3-7091-6242-2_9. URL http://dx.doi.org/10.1007/978-3-7091-6242-2_9. (Cited on page 130.)

Ramesh Raskar, Paul Beardsley, Jeroen van Baar, Yao Wang, Paul Dietz, Johnny Lee, Darren Leigh, and Thomas Willwacher. Rfig lamps: Interacting with a self-describing world via photosensing wireless tags and projectors. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6. doi: 10.1145/1185657.1185806. URL <http://doi.acm.org/10.1145/1185657.1185806>. (Cited on pages 171 and 175.)

Jun Rekimoto and Masanori Saitoh. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99,

- pages 378–385, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: 10.1145/302979.303113. URL <http://doi.acm.org/10.1145/302979.303113>. (Cited on pages 18, 20, and 219.)
- Julie Rico and Stephen Brewster. Gestures all around us: User differences in social acceptability perceptions of gesture based interfaces. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 64:1–64:2, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-281-8. doi: 10.1145/1613858.1613936. URL <http://doi.acm.org/10.1145/1613858.1613936>. (Cited on page 208.)
- Julie Rico and Stephen Brewster. Usable gestures for mobile interfaces: Evaluating social acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 887–896, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753458. URL <http://doi.acm.org/10.1145/1753326.1753458>. (Cited on page 208.)
- Jan Riemann, Mohammadreza Khalilbeigi, and Max Mühlhäuser. Peritop: extending back-projected tabletops with top-projected peripheral displays. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, ITS '13, pages 349–352, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512397. URL <http://doi.acm.org/10.1145/2512349.2512397>. (Cited on page 13.)
- Jan Riemann, Mohammadreza Khalilbeigi, Niloofar Dezfuli, and Max Mühlhäuser. Stacktop: Hybrid physical-digital stacking on interactive tabletops. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 1127–1132, New York, NY, USA, 2015a. ACM. ISBN 978-1-4503-3146-3. doi: 10.1145/2702613.2732742. URL <http://doi.acm.org/10.1145/2702613.2732742>. (Cited on page 13.)
- Jan Riemann, Mohammadreza Khalilbeigi, and Max Mühlhäuser. In-situ occlusion resolution for hybrid tabletop environments. In Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2015*, pages 278–295, Cham, Switzerland, 2015b. Springer International Publishing. ISBN 978-3-319-22698-9. URL http://dx.doi.org/10.1007/978-3-319-22698-9_18. (Cited on page 13.)

- Jan Riemann, Mohammadreza Khalilbeigi, Martin Schmitz, Sebastian Doeweling, Florian Müller, and Max Mühlhäuser. Freetop: Finding free spots for projective augmentation. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 1598–1606, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4082-3. doi: 10.1145/2851581.2892321. URL <http://doi.acm.org/10.1145/2851581.2892321>. (Cited on page 13.)
- Jan Riemann, Florian Müller, Sebastian Günther, and Max Mühlhäuser. An evaluation of hybrid stacking on interactive tabletops. In *Proceedings of the 2017 ACM Workshop on Interacting with Smart Objects*, SmartObject '17, pages 13–20, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4902-4. doi: 10.1145/3038450.3038451. URL <http://doi.acm.org/10.1145/3038450.3038451>. (Cited on page 13.)
- Jan Riemann, Markus Funk, Martin Schmitz, Christian Meurisch, and Max Mühlhäuser. Overtop: Breaking the boundaries of tangible tabletop environments. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, pages 231–234, New York, NY, USA, 2018a. ACM. ISBN 978-1-4503-5966-5. doi: 10.1145/3267305.3267559. URL <http://doi.acm.org/10.1145/3267305.3267559>. (Cited on page 13.)
- Jan Riemann, Martin Schmitz, Alexander Hendrich, and Max Mühlhäuser. Flowput: Environment-aware interactivity for tangible 3d objects. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1):31:1–31:23, March 2018b. ISSN 2474-9567. doi: 10.1145/3191763. URL <http://doi.acm.org/10.1145/3191763>. (Cited on page 13.)
- George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: Using spatial memory for document management. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 153–162, New York, NY, USA, 1998. ACM. ISBN 1-58113-034-1. doi: 10.1145/288392.288596. URL <http://doi.acm.org/10.1145/288392.288596>. (Cited on pages 23, 25, and 169.)
- Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *In Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011. (Cited on page 153.)

- Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation, ICRA'09*, pages 1848–1853, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2788-8. URL <http://dl.acm.org/citation.cfm?id=1703435.1703733>. (Cited on pages 132, 133, and 139.)
- Stephen Ruth and Imran Chaudhry. Telework: A productivity paradox? *IEEE Internet Computing*, 12(6):87–90, 2008. ISSN 1089-7801. doi: <http://doi.ieeecomputersociety.org/10.1109/MIC.2008.132>. (Cited on pages 9, 166, and 169.)
- M. Sano, K. Matsumoto, B. H. Thomas, and H. Saito. Rubix: Dynamic spatial augmented reality by extraction of plane regions with a rgb-d camera. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, pages 148–151, Sept 2015. doi: 10.1109/ISMAR.2015.43. (Cited on page 132.)
- Makoto Sato and Kaori Fujinami. Nonoverlapped view management for augmented reality by tabletop projection. *J. Vis. Lang. Comput.*, 25(6):891–902, December 2014. ISSN 1045-926X. doi: 10.1016/j.jvlc.2014.10.030. URL <http://dx.doi.org/10.1016/j.jvlc.2014.10.030>. (Cited on pages 109 and 112.)
- Munehiko Sato, Ivan Poupyrev, and Chris Harrison. Touché: Enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 483–492, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207743. URL <http://doi.acm.org/10.1145/2207676.2207743>. (Cited on pages 7, 127, and 129.)
- Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. Midas: Fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12*, pages 579–588, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1580-7. doi: 10.1145/2380116.2380189. URL <http://doi.acm.org/10.1145/2380116.2380189>. (Cited on pages 127 and 129.)
- Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. A series of tubes: Adding interactivity to 3d prints using

internal pipes. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 3–12, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3069-5. doi: 10.1145/2642918.2647374. URL <http://doi.acm.org/10.1145/2642918.2647374>. (Cited on pages 127 and 129.)

Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. Capricate: A fabrication pipeline to design and 3d print capacitive touch sensors for interactive objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 253–258, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3779-3. doi: 10.1145/2807442.2807503. URL <http://doi.acm.org/10.1145/2807442.2807503>. (Cited on pages 7 and 125.)

Martin Schmitz, Andreas Leister, Niloofar Dezfuli, Jan Riemann, Florian Müller, and Max Mühlhäuser. Liquido: Embedding liquids into 3d printed objects to sense tilting and motion. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 2688–2696, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4082-3. doi: 10.1145/2851581.2892275. URL <http://doi.acm.org/10.1145/2851581.2892275>. (Cited on page 13.)

Stacey D. Scott, M. Sheelagh T. Carpendale, and Stefan Habelski. Storage bins: Mobile storage for collaborative tabletop displays. *IEEE Comput. Graph. Appl.*, 25(4):58–65, July 2005. ISSN 0272-1716. doi: 10.1109/MCG.2005.86. URL <http://dx.doi.org/10.1109/MCG.2005.86>. (Cited on pages 83 and 85.)

Abigail Sellen and Richard Harper. Paper as an analytic resource for the design of new technologies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 319–326, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258780. URL <http://doi.acm.org/10.1145/258549.258780>. (Cited on pages 3 and 6.)

Abigail J. Sellen and Richard H.R. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003. ISBN 026269283X. (Cited on pages 17 and 169.)

- Thitirat Siriborvornratanakul and Masanori Sugimoto. Clutter-aware adaptive projection inside a dynamic environment. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pages 241–242, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-951-7. doi: 10.1145/1450579.1450633. URL <http://doi.acm.org/10.1145/1450579.1450633>. (Cited on pages 108 and 109.)
- Hyunyoung Song, Tovi Grossman, George Fitzmaurice, François Guimbretiere, Azam Khan, Ramtin Attar, and Gordon Kurtenbach. Penlight: Combining a mobile projector and a digital pen for dynamic visual overlay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 143–152, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518726. URL <http://doi.acm.org/10.1145/1518701.1518726>. (Cited on pages 17 and 20.)
- D. Spelmezan, R. M. González, and S. Subramanian. Skinhaptics: Ultrasound focused in the hand creates tactile sensations. In *2016 IEEE Haptics Symposium (HAPTICS)*, pages 98–105, Washington, DC, USA, April 2016. IEEE Computer Society. doi: 10.1109/HAPTICS.2016.7463162. (Cited on page 208.)
- Martin Spindler. Spatially aware tangible display interaction in a tabletop environment. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 277–282, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396679. URL <http://doi.acm.org/10.1145/2396636.2396679>. (Cited on pages 19 and 20.)
- Martin Spindler, Sophie Stellmach, and Raimund Dachsel. Paperlens. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS 09*, page 69, New York, NY, USA, 2009. ACM. ISBN 9781605587332. doi: 10.1145/1731903.1731920. URL <http://dx.doi.org/10.1145/1731903.1731920>. (Cited on pages 19 and 20.)
- Martin Spindler, Marcel Martsch, and Raimund Dachsel. Going beyond the surface: Studying multi-layer interaction above the tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1277–1286, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208583. URL <http://doi.acm.org/10.1145/2207676.2208583>. (Cited on page 19.)

- J. Steimle, O. Brdiczka, and M. Muhlhauser. Coscribe: Integrating paper and digital documents for collaborative knowledge work. *IEEE Transactions on Learning Technologies*, 2(3):174–188, July 2009. ISSN 1939-1382. doi: 10.1109/TLT.2009.27. (Cited on page 88.)
- Jürgen Steimle, Mohammadreza Khalilbeigi, and Max Mühlhäuser. Hybrid groups of printed and digital documents on tabletops: A study. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3271–3276, New York, NY, USA, 2010a. ACM. ISBN 978-1-60558-930-5. doi: 10.1145/1753846.1753970. URL <http://doi.acm.org/10.1145/1753846.1753970>. (Cited on page 28.)
- Jürgen Steimle, Mohammadreza Khalilbeigi, Max Mühlhäuser, and James D. Hollan. Physical and digital media usage patterns on interactive tabletop surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 167–176, New York, NY, USA, 2010b. ACM. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936685. URL <http://doi.acm.org/10.1145/1936652.1936685>. (Cited on pages 6, 28, 38, 54, 83, and 109.)
- Jürgen Steimle, Andreas Jordt, and Pattie Maes. Flexpad: Highly flexible bending interactions for projected handheld displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 237–246, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2470688. URL <http://doi.acm.org/10.1145/2470654.2470688>. (Cited on pages 110, 132, and 133.)
- Noi Sukaviriya, Rick Kjeldsen, Claudio Pinhanez, Lijun Tang, Anthony Levas, Gopal Pingali, and Mark Podlaseck. A portable system for anywhere interactions. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 789–790, New York, NY, USA, 2004. ACM. ISBN 1-58113-703-6. doi: 10.1145/985921.985937. URL <http://doi.acm.org/10.1145/985921.985937>. (Cited on pages 171 and 175.)
- Aurélien Tabard, Juan David Hincapié Ramos, and Jakob Bardram. The elabbench in the wild: Supporting exploration in a molecular biology lab. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 3051–3060, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208718. URL <http://doi.acm.org/10.1145/2207676.2208718>.

[//doi.acm.org/10.1145/2207676.2208718](https://doi.acm.org/10.1145/2207676.2208718). (Cited on pages 19, 20, and 57.)

Aurelien Tabard, Simon Gurn, Andreas Butz, and Jakob Bardram. A case study of object and occlusion management on the elabbench, a mixed physical/digital tabletop. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 251–254, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512794. URL <http://doi.acm.org/10.1145/2512349.2512794>. (Cited on pages 6 and 54.)

K. Tanaka, Y. Kishino, M. Miyamae, T. Terada, and S. Nishio. An information layout method for an optical see-through hmd considering the background. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 109–110, Washington, DC, USA, Oct 2007. IEEE Computer Society. doi: 10.1109/ISWC.2007.4373791. (Cited on page 114.)

Lucia Terrenghi, David Kirk, Hendrik Richter, Sebastian Krämer, Otmar Hilliges, and Andreas Butz. Physical handles at the interactive surface: Exploring tangibility and its benefits. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 138–145, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-141-5. doi: 10.1145/1385569.1385593. URL <http://doi.acm.org/10.1145/1385569.1385593>. (Cited on page 7.)

Vineet Thanedar and Tobias Höllerer. Semi-automated placement of annotations in videos. Technical Report 2004-11, UC, Santa Barbara, 2004. (Cited on page 114.)

A. Toney and B. H. Thomas. Considering reach in tangible and table top design. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06)*, pages 2 pp.–, Jan 2006. doi: 10.1109/TABLETOP.2006.9. (Cited on page 25.)

Kazuna Tsuboi, Yuji Oyamada, Maki Sugimoto, and Hideo Saito. 3d object surface tracking using partial shape templates trained from a depth camera for spatial augmented reality environments. In *Proceedings of the Fourteenth Australasian User Interface Conference - Volume 139*, AUIC '13, pages 125–126, Darlinghurst, Australia, Australia, 2013. Australian Computer Society, Inc. ISBN 978-1-921770-24-1. URL <http://dl.acm.org/citation.cfm?id=2525493.2525508>. (Cited on pages 132, 133, and 139.)

- Md. Sami Uddin, Carl Gutwin, and Andy Cockburn. The effects of artificial landmarks on learning and performance in spatial-memory interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3843–3855, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025497. URL <http://doi.acm.org/10.1145/3025453.3025497>. (Cited on page 179.)
- Brygg Ullmer and Hiroshi Ishii. The metadesk: Models and prototypes for tangible user interfaces. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 223–232, New York, NY, USA, 1997. ACM. ISBN 0-89791-881-9. doi: 10.1145/263407.263551. URL <http://doi.acm.org/10.1145/263407.263551>. (Cited on page 131.)
- John Underkoffler and Hiroshi Ishii. Urp: A luminous-tangible workbench for urban planning and design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 386–393, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: 10.1145/302979.303114. URL <http://doi.acm.org/10.1145/302979.303114>. (Cited on pages 110 and 112.)
- Craig Villamor, Dan Willis, Luke Wroblewski, and Jennifer Rhim. Touch gesture reference guide, April 2010. URL <http://www.lukew.com/ff/entry.asp?1071>. (Cited on page 86.)
- Daniel Vogel and Ravin Balakrishnan. Occlusion-aware interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 263–272, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753365. URL <http://doi.acm.org/10.1145/1753326.1753365>. (Cited on page 55.)
- Daniel Vogel and G ry Casiez. Hand occlusion on a multi-touch tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2307–2316, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208390. URL <http://doi.acm.org/10.1145/2207676.2208390>. (Cited on pages 55 and 57.)
- Daniel Vogel, Matthew Cudmore, G ry Casiez, Ravin Balakrishnan, and Liam Keliher. Hand occlusion with tablet-sized direct pen input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09,

- pages 557–566, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518787. URL <http://doi.acm.org/10.1145/1518701.1518787>. (Cited on page 55.)
- Manuela Waldner, Markus Steinberger, Raphael Grasset, and Dieter Schmalstieg. Importance-driven compositing window management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 959–968, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979085. URL <http://doi.acm.org/10.1145/1978942.1979085>. (Cited on pages 53 and 57.)
- Nayuko Watanabe, Motoi Washida, and Takeo Igarashi. Bubble clusters: An interface for manipulating spatial aggregation of graphical objects. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 173–182, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-679-0. doi: 10.1145/1294211.1294241. URL <http://doi.acm.org/10.1145/1294211.1294241>. (Cited on pages 83 and 85.)
- Pierre Wellner. The digitaldesk calculator: Tangible manipulation on a desktop display. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*, UIST '91, pages 27–33, New York, NY, USA, 1991. ACM. ISBN 0-89791-451-1. doi: 10.1145/120782.120785. URL <http://doi.acm.org/10.1145/120782.120785>. (Cited on pages 128, 129, and 133.)
- Pierre Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36(7):87–96, July 1993. ISSN 0001-0782. doi: 10.1145/159544.159630. URL <http://doi.acm.org/10.1145/159544.159630>. (Cited on pages 17, 20, 32, 80, 106, 131, and 218.)
- Steve Whittaker and Julia Hirschberg. The character, value, and management of personal paper archives. *ACM Trans. Comput.-Hum. Interact.*, 8(2):150–170, June 2001. ISSN 1073-0516. doi: 10.1145/376929.376932. URL <http://doi.acm.org/10.1145/376929.376932>. (Cited on page 3.)
- Daniel Wigdor, Darren Leigh, Clifton Forlines, Samuel Shipman, John Barnwell, Ravin Balakrishnan, and Chia Shen. Under the table interaction. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 259–268, New York, NY, USA, 2006.

ACM. ISBN 1-59593-313-1. doi: 10 . 1145 / 1166253 . 1166294. URL <http://doi.acm.org/10.1145/1166253.1166294>. (Cited on pages 33, 55, and 57.)

Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 269–278, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-679-0. doi: 10.1145/1294211.1294259. URL <http://doi.acm.org/10.1145/1294211.1294259>. (Cited on page 55.)

Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. Printed optics: 3d printing of embedded optical elements for interactive devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 589–598, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1580-7. doi: 10 . 1145 / 2380116 . 2380190. URL <http://doi.acm.org/10.1145/2380116.2380190>. (Cited on pages 7, 125, and 127.)

Karl D. D. Willis, Takaaki Shiratori, and Moshe Mahler. Hideout: Mobile projector interaction with tangible objects and surfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, pages 331–338, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1898-3. doi: 10 . 1145 / 2460625 . 2460682. URL <http://doi.acm.org/10.1145/2460625.2460682>. (Cited on page 43.)

A. D. Wilson. Depth-sensing video cameras for 3d tangible tabletop interaction. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, pages 201–204, Oct 2007. ISBN 0-7695-3013-3. doi: 10.1109/TABLETOP.2007.35. (Cited on page 110.)

Andrew D. Wilson. Playanywhere: A compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 83–92, New York, NY, USA, 2005. ACM. ISBN 1-59593-271-2. doi: 10.1145/1095034.1095047. URL <http://doi.acm.org/10.1145/1095034.1095047>. (Cited on pages 110, 128, 171, and 175.)

Andrew D. Wilson. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10,

- pages 69–72, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936665. URL <http://doi.acm.org/10.1145/1936652.1936665>. (Cited on pages 32, 128, and 129.)
- Robert Xiao, Chris Harrison, and Scott E. Hudson. Worldkit: Rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 879–888, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466113. URL <http://doi.acm.org/10.1145/2470654.2466113>. (Cited on pages 108, 109, 112, 128, and 129.)
- Robert Xiao, Scott Hudson, and Chris Harrison. Direct: Making touch tracking on ordinary surfaces practical with hybrid depth-infrared sensing. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, ISS '16, pages 85–94, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4248-3. doi: 10.1145/2992154.2992173. URL <http://doi.acm.org/10.1145/2992154.2992173>. (Cited on page 128.)
- Yang Zhang, Gierad Laput, and Chris Harrison. Electrick: Low-cost touch sensing using electric field tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1–14, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025842. URL <http://doi.acm.org/10.1145/3025453.3025842>. (Cited on page 127.)
- Zhengyou Zhang. Estimating projective transformation matrix (collineation, homography). Technical report, Microsoft Research, 1993/2010. (Cited on pages 44 and 45.)
- Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, Oct 1994. ISSN 1573-1405. doi: 10.1007/BF01427149. URL <https://doi.org/10.1007/BF01427149>. (Cited on page 140.)
- Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, November 2000. ISSN 0162-8828. doi: 10.1109/34.888718. URL <http://dx.doi.org/10.1109/34.888718>. (Cited on page 130.)
- Yi Zhou, Shuangjiu Xiao, Ning Tang, Zhiyong Wei, and Xu Chen. Pmomo: Projection mapping on movable 3d object. In *Proceedings of the 2016*

CHI Conference on Human Factors in Computing Systems, CHI '16, pages 781–790, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858329. URL <http://doi.acm.org/10.1145/2858036.2858329>. (Cited on page 132.)